EXPLAINING MODEL DECISIONS AND FIXING THEM VIA HUMAN FEEDBACK

A Dissertation
Presented to
The Academic Faculty

By

Ramprasaath R. Selvaraju

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology

May 2020

EXPLAINING MODEL DECISIONS AND FIXING THEM VIA HUMAN FEEDBACK

Approved by:

Dr. Devi Parikh, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Dhruv Batra
School of Interactive Computing
Georgia Institute of Technology

Dr. Judy Hoffman
School of Interactive Computing
Georgia Institute of Technology

Dr. Stefan Lee
School of Electrical Engineering and Computer Science
Oregon State University

Dr. Been Kim

Google Research

Date Approved: March 23, 2020

ACKNOWLEDGEMENTS

I am extremely grateful to my advisor, Devi Parikh, without whom none of this would have been possible. I would like to thank her for providing me with the opportunity to do an internship at her lab during my undergrad, and then for accepting me as a PhD student in her fantastic lab the following semester. It has been my privilege to work with Devi ever since. She is the best and the nicest advisor anyone could hope to have. Her emphasis on clarity of thought has shaped me immensely. I find her organizational skills extremely inspiring and I hope to inculcate these qualities not only in my research work but also in my life.

I would like to thank Dhruv Batra as well for his support throughout the course of my PhD. Dhruv is the best teacher that I have ever had the opportunity to learn from – I will never get bored of re-watching his Machine Learning lecture videos. Dhruv's very honest feedback has helped shape and refined my thought process significantly.

I would like to thank my committee members Judy Hoffman, Stefan Lee, and Been Kim for the fruitful discussions and comments on my thesis and during my defense.

I would like to thank my mentors at Microsoft Research, Ece Kamar, Besmira Nushi, and Marco Ribeiro, for their wonderful guidance and for making my internship there fun and worthwhile. I would also like to thank Mohammed Elhoseiny and Tilak Sharma, Yilin Shen, and Andrej Karpathy for mentoring me and providing a great environment during my internship at Facebook, Samsung and Tesla respectively.

I would like to express my deep gratitude to Ramakrishna Vedantam, Prithvijit Chattopadhyay, Stefan Lee, Michael Cogswell, and Ashwin Kalyan. I have learned quite a lot collaborating with them. Thanks for staying with me during some testing times.

I am immensely thankful to my friends and well-wishers, Harsh Agrawal, Arjun Chandrasekaran, Yash Goyal, Aishwarya Agrawal, Jianwei Yang and Jiasen Lu for being there at critical times in my PhD career. I would specially like to thank my labmate, Abhishek

Das, for all the brainstorming sessions, unfiltered comments, constant support, help and for being a wonderful roommate and a great friend.

I am extremely thankful to Shrenik Lad for the wonderful times we had at Oxford University and importantly for introducing me to Devi Parikh, which changed my life. I would also like to thank Stanislaw Antol who has been the first contact in the lab for many of my labmates. Stan helped quite a bit with getting accustomed with the new lab environment, helping set up the infrastructure and has generally being a very nice person.

I would also like to thank each and every member of the Computer Vision Machine Learning Perception (CVMLP) Group for providing a wonderful atmosphere and a great culture during my PhD. This includes – Viraj Prabhu, Deshraj Yadav, Purva Tendulkar, Ayush Srivastava, Samyak Datta, Rishabh Jain, Clint Solomon, Erik Wijmans, Mohit Sharma, Sameer Dharur, Faruk Ahmed, Qing Sun, Xiao Lin, Senthil Purushwalkam, Peng Zhang, Karan Desai, Sanyam Agrawal, Aroma Mahendru, Latha Pemula, Zhile Ren, Arijit Ray, Khushi Gupta, Ahmed Osman, Akrit Mohapatra, Meera Hahn, Joanne Truong, Peter Anderson, Vishvak Murahari and Arjun Majumdar. The immense constructive feedback the group generally provides during group presentations helped me grow and restructure my thought process. I am deeply honored to have been a part of this CVMLP Group.

My last 2 years of my PhD would not have been the same if not for Purva Tendulkar. Purva is by far the most positive person I have had the pleasure to meet and work with. I thank her for the positive atmosphere she created around me, for helping me stay motivated throughout, for being cheerful, and for making me a very mature person. She has been a part of two of my only papers which have been accepted (in fact as oral) in the first attempt. I hope our collaboration (and the record) continues.

I would also like to thank Purva, Abhishek and Varun for all the amazing bonding times, the sweet memories you have given me, and for keeping me sane during this quarantine period.

I would like to thank my parents, Nirmala and Selvaraju, for all their sacrifices through-

out my life for the well-being of me and my brother, Varun, for understanding and encouraging us to pursue our dreams and for their love and constant support throughout our schooling.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Thesis Statement

Network's neuron importance can be exploited to interpret decisions, facilitate knowledge transfer, correct unwanted biases and encourage human-like reasoning in AI models.

SUMMARY

Deep networks have enabled unprecedented breakthroughs in a variety of computer vision tasks. While these models enable superior performance, their increasing complexity and lack of decomposability into individually intuitive components makes them hard to interpret. Consequently, when today's intelligent systems fail, they fail spectacularly disgracefully, giving no warning or explanation.

Towards the goal of making deep networks interpretable, trustworthy and unbiased, in this dissertation, I will present my work on building algorithms that provide explanations for decisions emanating from deep networks in order to —

1. understand/interpret why the model did what it did,
2. enable knowledge transfer between humans and AI,
3. correct unwanted biases learned by AI models, and
4. encourage human-like reasoning in AI.

CHAPTER 1
INTRODUCTION

AI systems incorporating deep networks can be dependable tools (or teammates) for decision makers when they help humans develop an appropriate level of trust. This trust involves the humans being able to predict when and how the system will succeed or fail. In order to be able to build trust in intelligent systems and move towards their meaningful integration into our everyday lives, we must build 'transparent' models that have the ability to explain *why they predict what they predict*.

Broadly speaking, this transparency and ability to explain can be useful at three different stages of Artificial Intelligence (AI) evolution. First, when AI is significantly weaker than humans and not yet reliably deployable (*e.g.* visual question answering [1]), the goal of transparency and explanations could be to identify the failure modes [2, 3], thereby helping researchers focus their efforts on the most fruitful research directions. Second, when AI is on par with humans and reliably deployable (*e.g.*, image classification [4] trained on sufficient data), the goal could be to establish appropriate trust and confidence in users. Third, when AI is significantly stronger than humans (*e.g.* chess or Go [5]), the goal of explanations could be in machine teaching [6] – *i.e.*, a machine teaching a human about how to make better decisions.

A broad goal in AI is to build systems that can accurately learn the function the developer wants it to learn. A purely example-driven learning paradigm does not necessarily incentivize models to learn the actual underlying function. A fundamental step in order to move to a future where models behave according to human specifications would require humans to better understand the inner workings of the model – i.e. their reasoning behind decisions, which we refer to as explanations. These explanations could aid humans in understanding what causes the mismatch between the actual and the learned function, thereby

helping humans provide targeted feedback to models. This feedback can help update/fix models in order to make them cater well according to user specification.

Towards this goal, in this dissertation we will develop techniques to explain decisions made by vision-based AI systems. We will then use these explanations to understand when models make decisions for reasons different from human decision makers and improve them as a step towards making them right for the right reasons.

1.1 Visual explanations

In any vision-based task, a popular mode of interpreting model decisions include visually highlighting portion of the raw data (*e.g.* input image) that most influenced the decision. We refer to these visualizations as Visual Explanations – heat maps or gradient maps visualizing the regions of input that are 'important' for predictions from deep neural networks. In Chapter 3, we develop a technique called Gradient-weighted Class Activation Mapping (Grad-CAM). Grad-CAM, uses the gradients of any target concept (say 'dog' in a classification network or a sequence of words in captioning network) flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. Grad-CAM is applicable to a wide variety of CNN model-families: (1) CNNs with fully-connected layers (*e.g.* VGG), (2) CNNs used for structured outputs (*e.g.* captioning), (3) CNNs used in tasks with multi-modal inputs (*e.g.* visual question answering) or reinforcement learning, all *without architectural changes or re-training*. We combine Grad-CAM with existing fine-grained visualizations to create a high-resolution class-discriminative visualization, Guided Grad-CAM, and apply it to image classification, image captioning, and visual question answering (VQA) models, including ResNet-based architectures, making them more transparent and explainable.

1.2 Facilitating knowledge transfer between Humans and AI through explanations

Current generation deep models are extremely data hungry, so the most effective approach involves feeding a lot of labeled data. However, collecting instance level annotations is expensive, private, and scarce in many applications. Hence it becomes important to find cheap and efficient ways to provide supervision to neural networks. One such highly informative (and relatively cheap) form of supervision is expert domain knowledge from humans. Towards this end we propose an approach to bake this domain knowledge into deep models through explanations.

Visual explanations through Grad-CAM help us understand what the network has already learned. This includes information such as which neurons along the path are important for the decision and what concepts the individual neurons learn. By utilizing the concepts learned by the semantic neurons of the network, our approach NIWT (Neuron Importance-aware Weight Transfer) provides a way to embed domain knowledge information from humans into semantic neurons of the network in-order to learn classifiers for novel classes.

In Chapter 4 we provide details of NIWT, and demonstrate its ability to achieve state-of-the-art performance on Zero-Shot Learning and show that by relying on grounding neuron-importances in semantic human interpretable domains, NIWT is automatically able to explain network decisions in the form of text and provide names to neurons, indicating what concept each neuron looks at in an image.

1.3 Leveraging visual explanations to make vision and language models more grounded

Today's state-of-the-art deep models, especially for vision and language tasks are known to rely heavily on superficial correlations in training data. As a result, these models are often biased by language priors, and do not make predictions sufficiently grounded in the image content. For example, image captioning models often generate phrases like "standing next

to a tree" when talking about giraffes because trees tend to co-occur in images of giraffes in the COCO train set, and VQA models blindly answer "yellow" when asked, "What color are the bananas?". This often becomes apparent when explanation modalities such as Grad-CAM are employed to assess the evidence that the models are basing their decisions on.

Using context or overly relying on priors for making decisions makes systems develop internal (incorrect) biases that don't help generalize to new data distributions. For example, learning that boat always lies surrounded by water or that traffic cones are always orange, will prevent the model from recognizing boats outside of water, and identifying traffic cones of different color. Hence it becomes extremely important to make models not only make right decisions but also look at relevant/appropriate regions.

In chapter 5 we extend our focus to use the insights gained from Grad-CAM to make models look at appropriate regions when making decisions. In this work, we propose a generic approach called Human Importance-aware Network Tuning (HINT) that effectively leverages human demonstrations to improve visual grounding. HINT encourages deep networks to be sensitive to the same input regions as humans. Our approach optimizes the alignment between human attention maps and gradient-based network importances – ensuring that models learn not just to look at but rather rely on visual concepts that humans found relevant for a task when making predictions. We apply HINT to Visual Question Answering and Image Captioning tasks, outperforming top approaches on splits that penalize over-reliance on language priors (VQA-CP) using human attention demonstrations for just 6% of the training data.

1.4 Using human explanations to evaluate and enforce compositional reasoning in models

While verifying and ensuring that models are looking at the right regions is sufficient for simple perception tasks, for more complex tasks, it becomes important to also check and

ensure that the models learn the right reasoning on top of these regions. In chapter 6 we address this problem in the context of visual question answering (VQA) task. Existing VQA datasets contain questions with varying levels of complexity. While the majority of questions in these datasets require *perception* for recognizing existence, properties, and spatial relationships of entities, a significant portion of questions pose challenges that correspond to *reasoning tasks* – tasks that can only be answered through a synthesis of perception and knowledge about the world, logic and / or reasoning. This distinction allows us to notice when existing VQA models have consistency issues – they answer the reasoning question correctly but fail on associated low-level perception questions. For example, in Figure 6.1, models answer the complex reasoning question "Is the banana ripe enough to eat?" correctly, but fail on the associated perception question "Are the bananas mostly green or yellow?" indicating that the model answered the reasoning question correctly but likely for the wrong reason. We quantify the extent to which this phenomenon occurs by creating a new Reasoning split of the VQA dataset and collecting VQA-Introspect, a new dataset currently consisting of 132K new perception questions which serve as sub questions corresponding to the set of perceptual tasks needed to effectively answer the complex reasoning questions in the Reasoning split. Additionally, we propose an approach called Sub-Question Importance-aware Network Tuning (SQuINT), which encourages the model to attend do the same parts of the image when answering the reasoning question and the perception sub questions. We show that SQuINT improves model consistency significantly, also marginally improving its performance on the Reasoning questions in VQA, while also displaying qualitatively better attention maps.

1.5 Contributions

For visual explanations,

- We introduced Grad-CAM, a class-discriminative localization technique that generates visual explanations for *any* CNN-based network without requiring architectural changes or re-training. We evaluate Grad-CAM for localization (Section 3.3.1), and faithfulness to model (Section 3.4.3), where it outperforms baselines.
- We apply Grad-CAM to existing top-performing classification, captioning (Section 3.7.1), and VQA (Section 3.7.2) models. For image classification, our visualizations lend insight into failures of current CNNs (Section 3.5.1), showing that seemingly unreasonable predictions have reasonable explanations. For captioning and VQA, our visualizations expose that common CNN + LSTM models are often surprisingly good at localizing discriminative image regions despite not being trained on grounded image-text pairs.
- We show a proof-of-concept of how interpretable Grad-CAM visualizations help in diagnosing failure modes by uncovering biases in datasets. This is important not just for generalization, but also for fair and bias-free outcomes as more and more decisions are made by algorithms in society.
- We conduct human studies (Section 5.7) that show Guided Grad-CAM explanations are class-discriminative and not only help humans establish trust, but also help untrained users successfully discern a 'stronger' network from a 'weaker' one, *even when both make identical predictions.*

For facilitating knowledge transfer between humans and AI,

- We introduce a zero-short learning approach based on mapping unseen class descriptions to neural importance within a deep network and then optimizing unseen classifier weights to effectively combine these concepts. In contrast to existing approaches, our method is capable of explaining its zero-shot predictions with human-interpretable semantics from attributes or captions.

- We demonstrate the effectiveness of our approach by reporting state-of-the-art results on generalized zero-shot learning on CUB and AWA2 without altering the classifier weights for the 'seen' classes. We also show our approach can handle arbitrary forms of domain knowledge including attributes and image captions for unseen classes.

- We show how inverse mappings from neuron importance to domain knowledge can also be learned to provide interpretable explanations for the decisions made by newly learned classifiers for unseen classes.

For leveraging explanations to make vision and language models more grounded,

- We introduce Human Importance-aware Network Tuning (HINT), a general approach for constraining the sensitivity of deep networks to specific input regions and demonstrate that it significantly improves visual grounding for two vision and language tasks.
- We set a new state-of-the-art on the bias-sensitive VQA Under Changing Priors (VQA-CP) dataset [7].
- We conduct studies showing that humans find HINTed models more trustworthy than standard models.

For evaluating and enforcing human-like reasoning in VQA models,

- We propose a new split of the VQA dataset, containing only Reasoning questions that require common-sense reasoning beyond perception.
- For questions in the Reasoning split, we introduce VQA-Introspect, a new dataset currently consisting of 132k associated Perception sub-questions which humans perceive as containing the components needed to answer the original questions.
- We evaluate state-of-the-art VQA models on VQA-Introspect and find that they have consistency issues – they answer the reasoning question correctly but fail on associated low-level perception questions.

- We introduce SQuINT – a generic modeling approach that is inspired by the compositional learning paradigm observed in humans.

1.6 List of Publications

1. **Selvaraju, R. R.**, Tendulkar, P., Parikh, D., Horvitz, E., Ribeiro, M., Nushi, B., & Kamar, E. . SQuINTing at VQA Models: Introspecting VQA Models with Sub-Questions. To appear at Computer Vision and Pattern Recognition (CVPR), 2020.

2. **Selvaraju, R. R.**, Lee, S., Shen, Y., Jia, H., Ghosh, S., Heck, L., Batra, D. & Parikh, D. . Taking a HINT: Leveraging Explanations to Make Vision & Language Models More Grounded. In International Conference on Computer Vision (ICCV), 2019.

3. Tendulkar, P., Krishna, K. **Selvaraju, R. R.**, & Parikh, D. . Trick or TReAT: Thematic Reinforcement for Artistic Typography. In International Conference on Computational Creativity (ICCC), 2019.

4. **Selvaraju, R. R.**, Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. . Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In International Journal of Computer Vision (IJCV), 2019.

5. **Selvaraju, R. R.**, Chattopadhyay, P., Elhoseiny, M., Sharma, T., Batra, D., Parikh, D., & Lee, S. . Choose your Neuron: Incorporating Domain knowledge into Deep Networks. In European Conference on Computer Vision (ECCV), 2018.

6. **Selvaraju, R. R.**, Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. . Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In International Conference on Computer Vision (ICCV), 2017.

7. Vijayakumar, A. K., Cogswell, M., **Selvaraju, R. R.**, Sun, Q., Lee, S., Crandall, D., & Batra, D. . Diverse beam search: Decoding diverse solutions from neural sequence models. In AAAI conference on Artificial Intelligence (AAAI), 2017.

8. Chattopadhyay, P., Vedantam, R., **Selvaraju, R. R.**, Batra, D., & Parikh, D. . Counting Everyday Objects in Everyday Scenes. In Computer Vision and Pattern Recognition (CVPR) 2017.

CHAPTER 2
SITUATING THE WORK

In this chapter, we first discuss how our work on visual explanation relates to other research efforts in similar directions. We will then briefly introduce previous approaches to tackle the task of zero-shot Learning. Finally, we will discuss works that focus on improving the visual and textual reasoning abilities of vision and language models.

2.1 Visual Explanations

A number of previous works [8, 9, 10, 11] have visualized CNN predictions by highlighting 'important' pixels (*i.e*. change in intensities of these pixels have the most impact on the prediction score). A drawback of these approaches is that they are not class-discriminative as shown in chapter 3. Some visualization methods synthesize images to maximally activate a network unit [8, 12] or invert a latent representation [13, 14]. Although these can be high-resolution and class-discriminative. they are not specific to an input image and thus can't be used to explain a model's prediction at an instance level. Ribeiro *et al*. [15] and Fong *et al*. [16] use a secondary learning component in-order to explain decisions of deep models. Works such as Class Activation Mapping (CAM) [17] achieves interpretability by retraining a simplified architecture, and is applicable only to a particular kind of CNN architectures. However, contrary to these approaches, our gradient-based model-agnostic approach, Grad-CAM has the ability to obtain visual explanations for a wide variety of CNN-based model architectures without requiring any architectural changes or retraining or additional supervisory signal.

2.2 Zero-shot Learning

The zero-shot learning task requires recognizing object instances from previously unseen test categories. One long-pursued way to solve the this task is by leveraging knowledge about common attributes and shared parts (e.g., furry, striped, etc.). Earlier approaches (e.g., [18, 19, 20, 21]) model attributes as an intermediate layer that bridges the image features and class labels. Recent works have realized the limitation of the conditional independence assumption between image representation and class labels given the attributes [22, 23]. These methods usually model attributes in a continuous space with a core goal to learn a transformation between attributes to images. In contrast to these approaches, in Chapter 4 we directly map text-based domain knowledge (captions or attributes) to internal components (neurons) of deep neural networks rather than learning associative mappings between images and text – offering not only comparable performance but also interpretability in our novel classifiers.

2.3 Making vision and language models right for right reasons

Vision and Language tasks. Image Captioning [24] and Visual Question Answering (VQA) [1] have emerged as two of the most widely studied vision-and-language problems. The image captioning task requires generating natural language description of image contents and the VQA task requires answering free-from questions about images. In both, models must learn to associate image content with complex free-form text. Since current training protocols for VQA involve training on input-output pairs without providing grounding, when there exists easier to learn correlations in language, models tend to exploit them. Consequentially, attention based models that explicitly reason about image-text correspondences have become the dominant paradigm [25, 26, 27, 28, 29, 30]. There has been growing evidence that even well performing attention-based models [25, 26, 27, 28, 29, 30] still latch onto language biases and fail to answer questions for the right reasons [7,

31, 32, 33].

Reducing effect of language bias in Vision and Language Models Hendricks *et al*. [33] study the generation of gender-specific words in image captioning – showing that models nearly always associated male gendered words to people performing extreme sports like snowboarding regardless of the image content. Recently, Agrawal *et al*. introduced a new split of the VQA dataset, namely VQA-CP (VQA under Changing Priors) dataset, that is constructed making sure the distributions between the train and test splits are different. Consequentially, models that do not learn to ground their decisions or models which overly rely on context or language priors tend to perform poorly on this split. As shown by Agrawal *et al*., even state of the art models suffer an extreme drop in performance when trained and evaluated on the VQA-CP split. Goyal *et al*. [34] noted an inherent language bias in the VQAv1 [1] dataset that are easily exploited by deep models. Hence they collected complementary images for each question in the VQAv1 dataset, such that the answer to the question for the new image is different, thus creating a balanced VQA dataset. This makes it harder for models to exploit the language biases. This is a very expensive process to make vision and language models less biased. In contrast, our approach (in chapter 5 directly incorporates human supervision for visual grounding – forcing models to base their decisions on the same regions as human respondents. Rather than relying on collecting expensive annotation or creating novel splits, our approach uses annotations from existing datasets to improve visual grounding.

Datasets for human reasoning A variety of datasets have been released with attention annotations on the image pointing to regions that are important to answer questions ([35, 36]), with corresponding work on enforcing such grounding [37, 38, 31]. In Chapter 6 we provide language-based grounding (rather than visual) through perception sub-question answers, and further evaluate the link between perception components and how they are composed by models and try to enforce right reasoning during learning. Closer to our work is the dataset of Lisa et al. [36], where natural language justifications are associated

with (question, answer) pairs. However, most of the questions contemplated (like much of the VQA dataset) pertain to perception questions (e.g. for the question-answer "What is the person doing? Snowboarding", the justification is "...they are on a snowboard ..."). Furthermore, it is hard to use natural language justifications to evaluate models that do not generate similar rationales (i.e. most SOTA models), or even coming up with metrics for models that do. In contrast, our dataset and evaluation is in the same modality (QA) that models are already trained to handle.

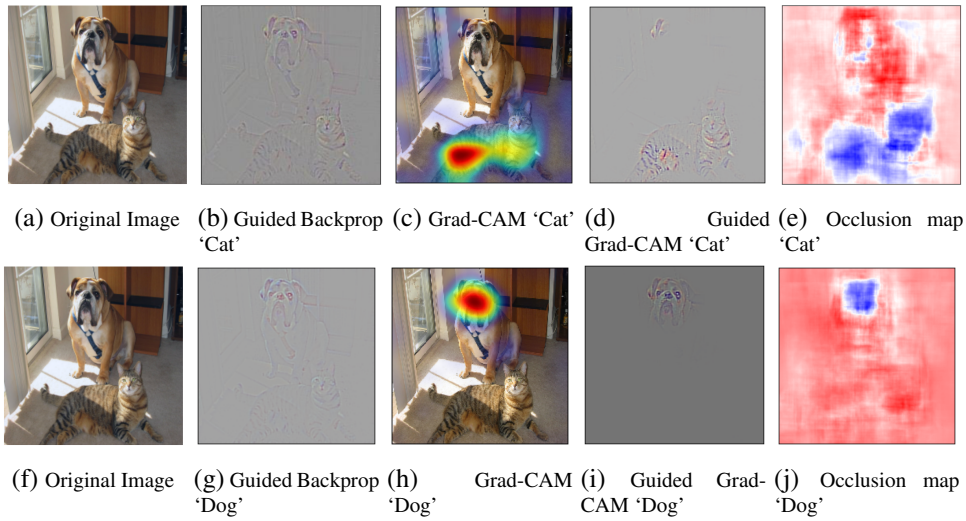
CHAPTER 3
GRAD-CAM: VISUAL EXPLANATIONS

3.1 Introduction

Deep neural models based on Convolutional Neural Networks (CNNs) have enabled unprecedented breakthroughs in a variety of computer vision tasks, from image classification [39, 40], object detection [41], semantic segmentation [42] to image captioning [43, 44, 45, 46], visual question answering [1, 47, 48, 49] and more recently, visual dialog [50, 51, 52] and embodied question answering [53, 54]. While these models enable superior performance, their lack of decomposability into *individually intuitive* components makes them hard to interpret [55]. Consequently, when today's intelligent systems fail, they often fail spectacularly disgracefully without warning or explanation, leaving a user staring at an incoherent output, wondering why the system did what it did.

There typically exists a trade-off between accuracy and simplicity or interpretability. Classical rule-based or expert systems [56] are highly interpretable but not very accurate (or robust). Decomposable pipelines where each stage is hand-designed are thought to be more interpretable as each individual component assumes a natural intuitive explanation. By using deep models, we sacrifice interpretable modules for uninterpretable ones that achieve greater performance through greater abstraction (more layers) and tighter integration (end-to-end training). Recently introduced deep residual networks (ResNets) [40] are over 200-layers deep and have shown state-of-the-art performance in several challenging tasks Such complexity makes these models hard to interpret. As such, deep models are beginning to explore the spectrum between interpretability and accuracy.

Zhou *et al*. [17] proposed a technique called Class Activation Mapping (CAM) for identifying discriminative regions used by a restricted class of image classification CNNs



(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat' (d) Guided Grad-CAM 'Cat' (e) Occlusion map 'Cat'

(f) Original Image (g) Guided Backprop 'Dog' (h) Grad-CAM 'Dog' (i) Guided Grad-CAM 'Dog' (j) Occlusion map 'Dog'

Figure 3.1: (a) Original image with a cat and a dog. (b-e) Support for the cat category according to various visualizations for VGG-16. (b) Guided Backpropagation [9]: highlights all contributing features. (c, g) Grad-CAM (Ours): localizes class-discriminative regions, (d) Combining (b) and (c) gives Guided Grad-CAM, which gives high-resolution class-discriminative visualizations. Interestingly, the localizations achieved by our Grad-CAM technique, (c) are very similar to results from occlusion sensitivity (e), while being orders of magnitude cheaper to compute. Note that in (c, h), red regions corresponds to high score for class, while in (e, j), blue corresponds to evidence for the class. Figure best viewed in color.

which do not contain any fully-connected layers. In essence, this work trades off model complexity and performance for more transparency into the working of the model. In contrast, we make existing state-of-the-art deep models interpretable without altering their architecture, thus avoiding the interpretability *vs*. accuracy trade-off. Our approach is a generalization of CAM [17] and is applicable to a significantly broader range of CNN model families: (1) CNNs with fully-connected layers (*e.g*. VGG), (2) CNNs used for structured outputs (*e.g*. captioning), (3) CNNs used in tasks with multi-modal inputs (*e.g*. VQA) or reinforcement learning, without requiring architectural changes or re-training.

What makes a good visual explanation? Consider image classification [57] – a 'good' visual explanation from the model for justifying any target category should be (a) class-discriminative (*i.e*. localize the category in the image) and (b) high-resolution (*i.e*. capture fine-grained detail).

Fig. 1 shows outputs from a number of visualizations for the 'tiger cat' class (top) and 'boxer' (dog) class (bottom). Pixel-space gradient visualizations such as Guided Backprop-

agation [9] and Deconvolution [10] are high-resolution and highlight fine-grained details in the image, but are not class-discriminative (Fig. 3.1b and Fig. 3.1g are very similar).

In contrast, localization approaches like CAM or our proposed method Gradient-weighted Class Activation Mapping (Grad-CAM), are highly class-discriminative (the 'cat' explanation exclusively highlights the 'cat' regions but not 'dog' regions in Fig. 3.1c, and viceversa in Fig. 3.1h).

In order to combine the best of both worlds, we show that it is possible to fuse existing pixel-space gradient visualizations with Grad-CAM to create Guided Grad-CAM visualizations that are both high-resolution and class-discriminative. As a result, important regions of the image which correspond to any decision of interest are visualized in high-resolution detail even if the image contains evidence for multiple possible concepts, as shown in Figures 1d and 1j. When visualized for 'tiger cat', Guided Grad-CAM not only highlights the cat regions, but also highlights the stripes on the cat, which is important for predicting that particular variety of cat.

To summarize, our contributions are as follows:

(1) We introduce Grad-CAM, a class-discriminative localization technique that generates visual explanations for *any* CNN-based network without requiring architectural changes or re-training. We evaluate Grad-CAM for localization (Section 3.3.1), and faithfulness to model (Section 3.4.3), where it outperforms baselines.

(2) We apply Grad-CAM to existing top-performing classification, captioning (Section 3.7.1), and VQA (Section 3.7.2) models. For image classification, our visualizations lend insight into failures of current CNNs (Section 3.5.1), showing that seemingly unreasonable predictions have reasonable explanations. For captioning and VQA, our visualizations expose that common CNN + LSTM models are often surprisingly good at localizing discriminative image regions despite not being trained on grounded image-text pairs.

(3) We show a proof-of-concept of how interpretable Grad-CAM visualizations help in diagnosing failure modes by uncovering biases in datasets. This is important not just for

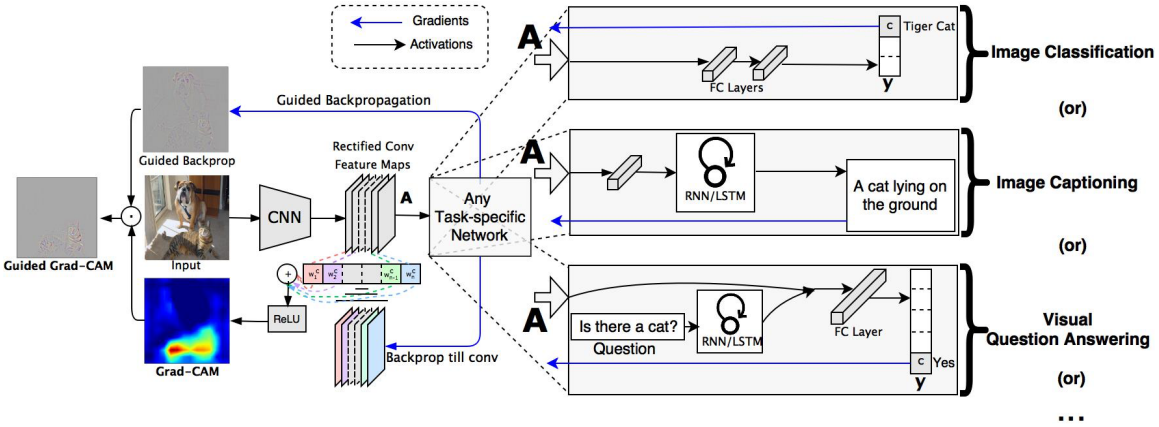


Figure 3.2: Grad-CAM overview: Given an image and a class of interest (*e.g.*, 'tiger cat' or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to computevvvvvv the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Finally, we pointwise multiply the heatmap with guided backpropagation to get Guided Grad-CAM visualizations which are both high-resolution and concept-specific.

generalization, but also for fair and bias-free outcomes as more and more decisions are made by algorithms in society.

(4) We present Grad-CAM visualizations for ResNets [40] applied to image classification and VQA (Section 3.7.2).

(5) We use neuron importance from Grad-CAM and neuron names from [58] and obtain textual explanations for model decisions (Section 3.6).

(6) We conduct human studies (Section 5.7) that show Guided Grad-CAM explanations are class-discriminative and not only help humans establish trust, but also help untrained users successfully discern a 'stronger' network from a 'weaker' one, *even when both make identical predictions.*

3.2 Grad-CAM

A number of previous works have asserted that deeper representations in a CNN capture higher-level visual constructs [59, 13]. Furthermore, convolutional layers naturally retain

spatial information which is lost in fully-connected layers, so we can expect the last convolutional layers to have the best compromise between high-level semantics and detailed spatial information. The neurons in these layers look for semantic class-specific information in the image (say object parts). Grad-CAM uses the gradient information flowing into the last convolutional layer of the CNN to assign importance values to each neuron for a particular decision of interest. Although our technique is fairly general in that it can be used to explain activations in any layer of a deep network, in this work, we focus on explaining output layer decisions only.

As shown in Fig. 3.2, in order to obtain the class-discriminative localization map Grad-CAM $L^c_{\text{Grad-CAM}} \in \mathbb{R}^{u \times v}$ of width u and height v for any class c, we first compute the gradient of the score for class c, y^c (before the softmax), with respect to feature map activations A^k of a convolutional layer, *i.e.* $\frac{\partial y^c}{\partial A^k}$. These gradients flowing back are global-average-pooled [1] over the width and height dimensions (indexed by i and j respectively) to obtain the neuron importance weights α^c_k:

$$\alpha^c_k = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A^k_{ij}}}_{\text{gradients via backprop}} \tag{3.1}$$

During computation of α^c_k while backpropagating gradients with respect to activations, the exact computation amounts to successive matrix products of the weight matrices and the gradient with respect to activation functions till the final convolution layer that the gradients are being propagated to. Hence, this weight α^c_k represents a *partial linearization* of the deep network downstream from A, and captures the 'importance' of feature map k for a target class c.

We perform a weighted combination of forward activation maps, and follow it by a

[1] Empirically we found global-average-pooling to work better than global-max-pooling as can be found in the Appendix.

ReLU to obtain,

$$L^c_{\text{Grad-CAM}} = ReLU \underbrace{\left(\sum_k \alpha^c_k A^k \right)}_{\text{linear combination}} \tag{3.2}$$

Notice that this results in a coarse heatmap of the same size as the convolutional feature maps (14×14 in the case of last convolutional layers of VGG [60] and AlexNet [39] networks) [2]. We apply a ReLU to the linear combination of maps because we are only interested in the features that have a *positive* influence on the class of interest, *i.e*. pixels whose intensity should be *increased* in order to increase y^c. Negative pixels are likely to belong to other categories in the image. As expected, without this ReLU, localization maps sometimes highlight more than just the desired class and perform worse at localization. Figures 1c, 1f and 1i, 1l show Grad-CAM visualizations for 'tiger cat' and 'boxer (dog)' respectively. Ablation studies are available in Section 3.8.

In general, y^c need not be the class score produced by an image classification CNN. It could be any differentiable activation including words from a caption or answer to a question. We provide qualitative results for Grad-CAM and Guided Grad-CAM applied to the task of image classification in Fig. 3.3. The results reported in Fig. 3.3 correspond to the VGG-16 [60] network trained on ImageNet.

Fig. 3.3 shows randomly sampled examples from COCO [24] validation set. COCO images typically have multiple objects per image and Grad-CAM visualizations show precise localization to support the model's prediction. Guided Grad-CAM can even localize tiny objects. For example our approach correctly localizes the predicted class "torch" (Fig. 3.3.a) inspite of its size and odd location in the image. Our method is also class-discriminative – it places attention *only* on the "toilet seat" even when a popular ImageNet category "dog" exists in the image (Fig. 3.3.e).

[2] We find that Grad-CAM maps become progressively worse as we move to earlier convolutional layers as they have smaller receptive fields and only focus on less semantic local features.

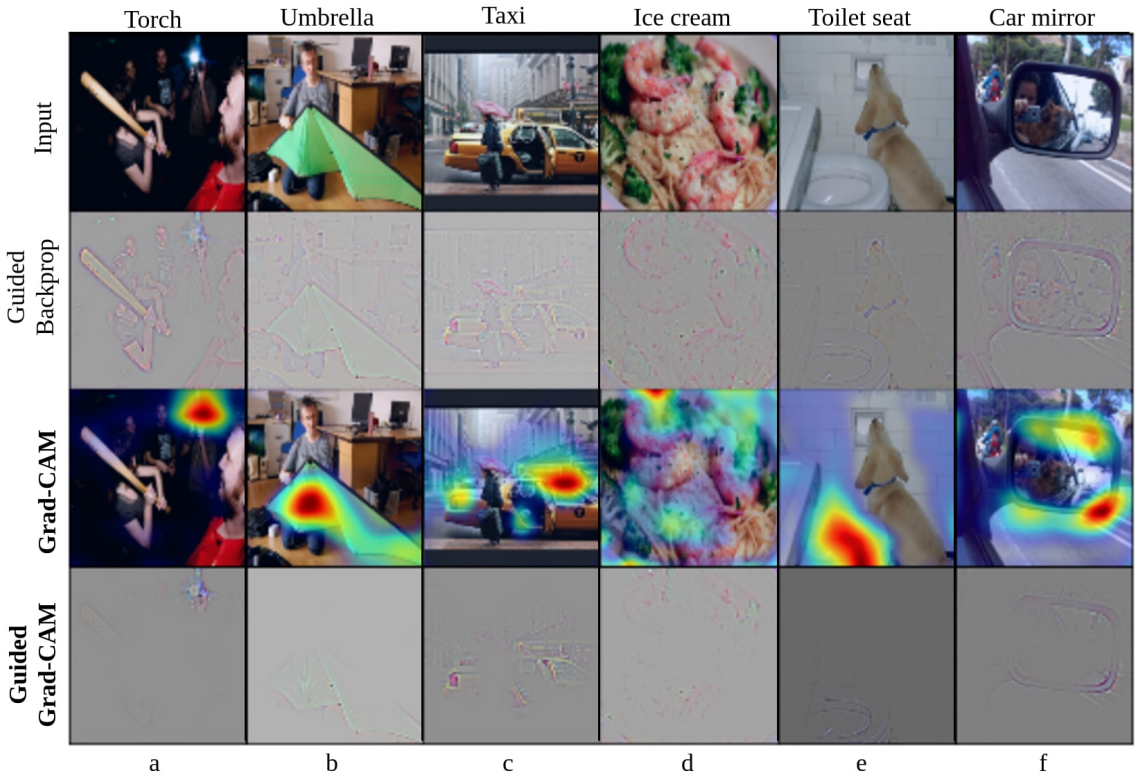


Figure 3.3: Visualizations for randomly sampled images from the COCO validation dataset. Predicted classes are mentioned at the top of each column.

3.2.1 Grad-CAM generalizes CAM

In this section, we discuss the connections between Grad-CAM and Class Activation Mapping (CAM) [17], and formally prove that Grad-CAM generalizes CAM for a wide variety of CNN-based architectures. Recall that CAM produces a localization map for an image classification CNN with a specific kind of architecture where global average pooled convolutional feature maps are fed directly into softmax. Specifically, let the penultimate layer produce K feature maps, $A^k \in \mathbb{R}^{u \times v}$, with each element indexed by i, j. So A^k_{ij} refers to the activation at location (i, j) of the feature map A^k. These feature maps are then spatially pooled using Global Average Pooling (GAP) and linearly transformed to produce a score Y^c for each class c,

$$Y^c = \sum_k \underbrace{w^c_k}_{\text{class feature weights}} \overbrace{\frac{1}{Z} \sum_i \sum_j}^{\text{global average pooling}} \underbrace{A^k_{ij}}_{\text{feature map}} \tag{3.3}$$

Let us define F^k to be the global average pooled output,

$$F^k = \frac{1}{Z} \sum_i \sum_j A^k_{ij} \tag{3.4}$$

CAM computes the final scores by,

$$Y^c = \sum_k w^c_k \cdot F^k \tag{3.5}$$

where w^c_k is the weight connecting the k^{th} feature map with the c^{th} class. Taking the gradient of the score for class c (Y^c) with respect to the feature map F^k we get,

$$\frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A^k_{ij}}}{\frac{\partial F^k}{\partial A^k_{ij}}} \tag{3.6}$$

Taking partial derivative of (3.4) w.r.t. A^k_{ij}, we can see that $\frac{\partial F^k}{\partial A^k_{ij}} = \frac{1}{Z}$. Substituting this

in (3.6), we get,

$$\frac{\partial Y^c}{\partial F^k} = \frac{\partial Y^c}{\partial A^k_{ij}} \cdot Z \tag{3.7}$$

From (3.5) we get that, $\frac{\partial Y^c}{\partial F^k} = w^c_k$. Hence,

$$w^c_k = Z \cdot \frac{\partial Y^c}{\partial A^k_{ij}} \tag{3.8}$$

Summing both sides of (3.8) over all pixels (i, j),

$$\sum_i \sum_j w^c_k = \sum_i \sum_j Z \cdot \frac{\partial Y^c}{\partial A^k_{ij}} \tag{3.9}$$

Since Z and w^c_k do not depend on (i, j), rewriting this as

$$Z w^c_k = Z \sum_i \sum_j \frac{\partial Y^c}{\partial A^k_{ij}} \tag{3.10}$$

Note that Z is the number of pixels in the feature map (or $Z = \sum_i \sum_j \mathbf{1}$). Thus, we can re-order terms and see that

$$w^c_k = \sum_i \sum_j \frac{\partial Y^c}{\partial A^k_{ij}} \tag{3.11}$$

Up to a proportionality constant $(1/Z)$ that gets normalized-out during visualization, the expression for w^c_k is identical to α^c_k used by Grad-CAM (3.1). Thus, Grad-CAM is a strict generalization of CAM. This generalization allows us to generate visual explanations from CNN-based models that cascade convolutional layers with much more complex interactions, such as those for image captioning and VQA (Sec. 3.7.2).

3.2.2 Grad-CAM is class-discriminative

In this section we show why Grad-CAM visualizations are class-discriminative. Recall that α^k_c (3.1) can be extracted from any layer of the deep CNN. Consider a simple cascaded deep CNN for classification having non-linear activation functions given by $\sigma_l(.)$ between layers l and $(l + 1)$. The scores corresponding to classes can be expressed as $\mathbf{y} = W^T_f A_f$

where W_f and A_f correspond to the final layer weights and activations respectively. Let o_c be the incoming gradient from the loss layer. Note that o_c is a one-hot vector with 1 at the dimension corresponding to class c and 0 everywhere else. With this, α_c^k of Grad-CAM can be written in terms of $\mathbf{y}$ and o_c as,

$$\alpha_c^k = \frac{1}{Z} \sum_i \sum_j \frac{\partial(\mathbf{y} \circ o_c)}{\partial A_{ij}^k} \tag{3.12}$$

where $\circ$ denotes the elementwise product. Since o_c is a constant, $\frac{\partial(o_c)}{\partial A_{ij}^k} = 0$. Thus we obtain,

$$\alpha_c^k = (\frac{1}{Z} \sum_i \sum_j \frac{\partial \mathbf{y}}{\partial A_{ij}^k}) \circ o_c \tag{3.13}$$

Now, recall that due to chain rule, this gradient expression above can be rewritten as a product of partial derivatives along the computation path:

$$\alpha_c^k = (\frac{1}{Z} \sum_{i,j} \frac{\partial \mathbf{y}}{\partial A_{f-1}} \frac{\partial A_{f-1}}{\partial \sigma_{A_{f-1}}} \frac{\partial \sigma_{A_{f-1}}}{\partial A_{f-2}} \frac{\partial A_{f-2}}{\partial \sigma_{A_{f-2}}} ...) \circ o_c \tag{3.14}$$

Recall that $\frac{\partial(y)}{\partial A_{f-1}}$ is W_f, and similarly $\frac{\partial \sigma_{A_l}}{\partial A_{l-1}}$ is W_l. The gradients w.r.t. activation functions results in diagonal matrices, D_{σ_l}. For networks with ReLU activations, the entries in the diagonal matrices are either 1 or 0 depending on whether the forward activations at that location is positive or negative, respectively. By substituting we get,

$$\alpha_c^k = (\frac{1}{Z} \sum_{i,j} W_f D_{\sigma_{f-1}} W_{f-1} D_{\sigma_{f-2}} ...) \circ o_c \tag{3.15}$$

The above expression can be reduced to,

$$\alpha_c^k = (\frac{1}{Z} \sum_{i,j} \prod_{l=f}^{L} D_{\sigma_{l-1}} W_l) \circ o_c \tag{3.16}$$

where W_l denote the weights connecting layers $(l-1)$ and l, and $l = \{f, f-1, ..., L\}$. Therefore, α_c^k explicitly captures the dynamics of the pathways leading from L (last convolutional layer) to the target class score, y_c in the network. This makes neuron-importance weights, α_c^k and by extension Grad-CAM class-discriminative.

3.2.3 Guided Grad-CAM

While Grad-CAM is class-discriminative and localizes relevant image regions, it lacks the ability to highlight fine-grained details like pixel-space gradient visualization methods (Guided Backpropagation [9], Deconvolution [10]). Guided Backpropagation visualizes gradients with respect to the image where negative gradients are suppressed when backpropagating through ReLU layers. Intuitively, this aims to capture pixels detected by neurons, not the ones that suppress neurons. See Figure 1c, where Grad-CAM can easily localize the cat; however, it is unclear from the coarse heatmap why the network predicts this particular instance as 'tiger cat'. In order to combine the best aspects of both, we fuse Guided Backpropagation and Grad-CAM visualizations via element-wise multiplication ($L^c_{\text{Grad-CAM}}$ is first upsampled to the input image resolution using bilinear interpolation). Fig. 3.2 bottom-left illustrates this fusion. This visualization is both high-resolution (when the class of interest is 'tiger cat', it identifies important 'tiger cat' features like stripes, pointy ears and eyes) and class-discriminative (it highlights the 'tiger cat' but not the 'boxer (dog)'). Replacing Guided Backpropagation with Deconvolution gives similar results, but we found Deconvolution visualizations to have artifacts and Guided Backpropagation to be generally less noisy.

3.2.4 Counterfactual Explanations

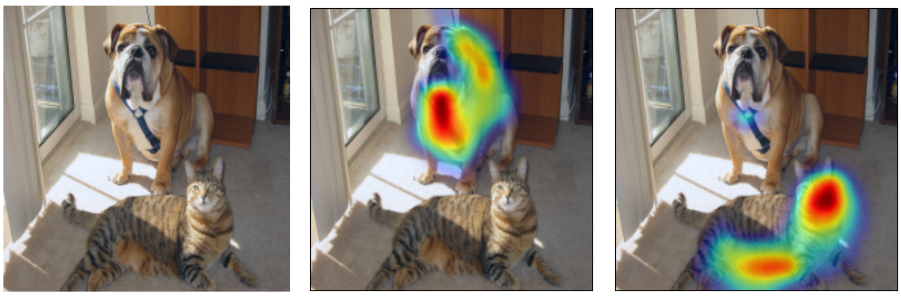
Using a slight modification to Grad-CAM, we can obtain explanations that highlight support for regions that would make the network change its prediction. As a consequence, removing concepts occurring in those regions would make the model more confident about

its prediction. We refer to this explanation modality as counterfactual explanations.

Specifically, we negate the gradient of y^c (score for class c) with respect to feature maps A of a convolutional layer. Thus the importance weights α_k^c now become

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{-\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{Negative gradients}} \tag{3.17}$$

As in (3.2), we take a weighted sum of the forward activation maps, A, with weights α_k^c, and follow it by a ReLU to obtain counterfactual explanations as shown in Fig. 3.4.



(a) Original Image (b) Cat Counterfactual exp (c) Dog Counterfactual exp

Figure 3.4: Counterfactual Explanations with Grad-CAM

3.3 Evaluating Localization Ability of Grad-CAM

3.3.1 Weakly-supervised Localization

In this section, we evaluate the localization capability of Grad-CAM in the context of image classification. The ImageNet localization challenge [57] requires approaches to provide bounding boxes in addition to classification labels. Similar to classification, evaluation is performed for both the top-1 and top-5 predicted categories.

Given an image, we first obtain class predictions from our network and then generate Grad-CAM maps for each of the predicted classes and binarize them with a threshold of 15% of the max intensity. This results in connected segments of pixels and we draw a bounding box around the single largest segment. Note that this is weakly-supervised

localization – the models were never exposed to bounding box annotations during training.

We evaluate Grad-CAM localization with off-the-shelf pretrained VGG-16 [60], AlexNet [39] and GoogleNet [61] (obtained from the Caffe [62] Zoo). Following ILSVRC-15 evaluation, we report both top-1 and top-5 localization errors on the val set in Table. 3.1. Grad-CAM localization errors are significantly better than those achieved by c-MWP [63] and Simonyan *et al*. [8], which use grab-cut to post-process image space gradients into heat maps. Grad-CAM for VGG-16 also achieves better top-1 localization error than CAM [17], which requires a change in the model architecture, necessitates re-training and thereby achieves worse classification errors (2.98% worse top-1), while Grad-CAM does not compromise on classification performance.

Table 3.1: Classification and localization error % on ILSVRC-15 val (lower is better) for VGG-16, AlexNet and GoogleNet. We see that Grad-CAM achieves superior localization errors without compromising on classification performance.

		Classification		**Localization**	
		Top-1	**Top-**5	**Top-**1	**Top-**5
VGG-16	Backprop [8]	30.38	10.89	61.12	51.46
	c-MWP [63]	30.38	10.89	70.92	63.04
	Grad-CAM (ours)	30.38	10.89	**56.51**	46.41
	CAM [17]	33.40	12.20	57.20	**45.14**
AlexNet	c-MWP [63]	44.2	20.8	92.6	89.2
	Grad-CAM (ours)	44.2	20.8	68.3	56.6
GoogleNet	Grad-CAM (ours)	31.9	11.3	60.09	49.34
	CAM [17]	31.9	11.3	60.09	49.34

3.3.2 Weakly-supervised Segmentation

Semantic segmentation involves the task of assigning each pixel in the image an object class (or background class). Being a challenging task, this requires expensive pixel-level

annotation. The task of weakly-supervised segmentation involves segmenting objects with just image-level annotation, which can be obtained relatively cheaply from image classification datasets. In recent work, Kolesnikov *et al*. [64] introduced a new loss function for training weakly-supervised image segmentation models. Their loss function is based on three principles – 1) to seed with weak localization cues, encouraging segmentation network to match these cues, 2) to expand object seeds to regions of reasonable size based on information about which classes can occur in an image, 3) to constrain segmentations to object boundaries that alleviates the problem of imprecise boundaries already at training time. They showed that their proposed loss function, consisting of the above three losses leads to better segmentation.

However, their algorithm is sensitive to the choice of weak localization seed, without which the network fails to localize objects correctly. In their work, they used CAM maps from a VGG-16 based network which are used as object seeds for weakly localizing foreground classes. We replaced the CAM maps with Grad-CAM obtained from a standard VGG-16 network and obtain a Intersection over Union (IoU) score of 49.6 (compared to 44.6 obtained with CAM) on the PASCAL VOC 2012 segmentation task. Fig. 3.5 shows some qualitative results. More examples are available in [65].

3.3.3 Pointing Game

Zhang *et al*. [63] introduced the Pointing Game experiment to evaluate the discriminativeness of different visualization methods for localizing target objects in scenes. Their evaluation protocol first cues each visualization technique with the ground-truth object label and extracts the maximally activated point on the generated heatmap. It then evaluates if the point lies within one of the annotated instances of the target object category, thereby counting it as a hit or a miss.

The localization accuracy is then calculated as
$Acc = \frac{\#Hits}{\#Hits+\#Misses}$. However, this evaluation only measures precision of the visualiza-

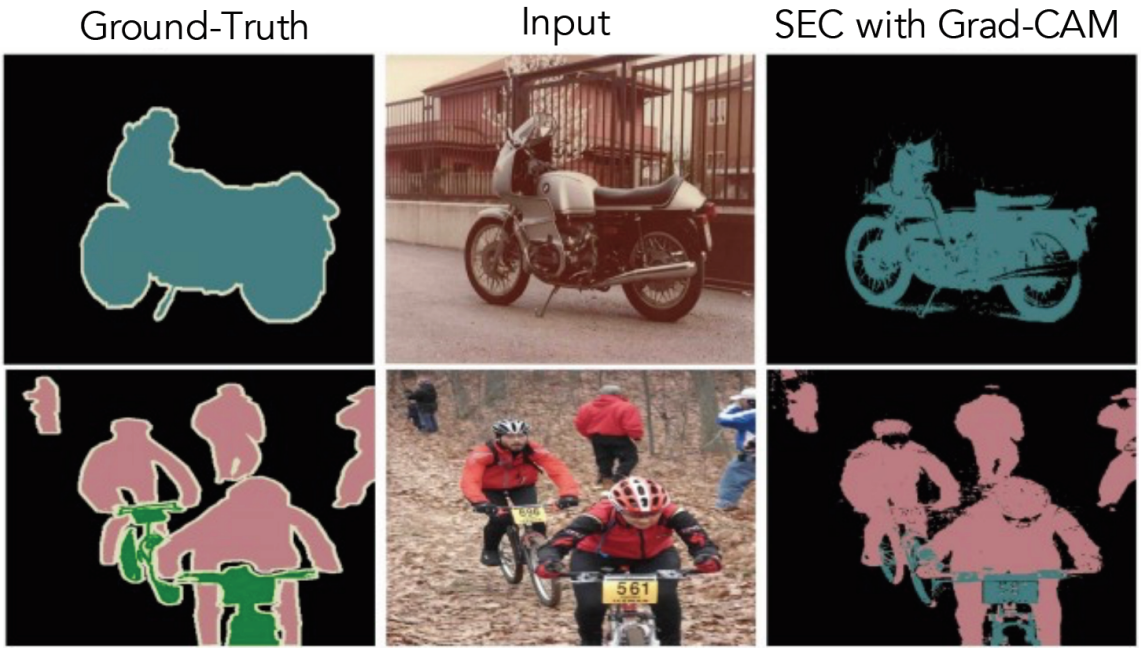


Figure 3.5: PASCAL VOC 2012 Segmentation results with Grad-CAM as seed for SEC [64].

tion technique. We modify the protocol to also measure recall – we compute localization maps for top-5 class predictions from the CNN classifiers[3] and evaluate them using the pointing game setup with an additional option to reject any of the top-5 predictions from the model if the maximally activated point in the map is below a threshold, *i.e*. if the visualization correctly rejects the predictions which are absent from the ground-truth categories, it gets that as a hit. We find that Grad-CAM outperforms c-MWP [63] by a significant margin (70.58% *vs*. 60.30%). Qualitative examples comparing c-MWP [63] and Grad-CAM on can be found in Section A.5[4].

3.4 Evaluating Visualizations

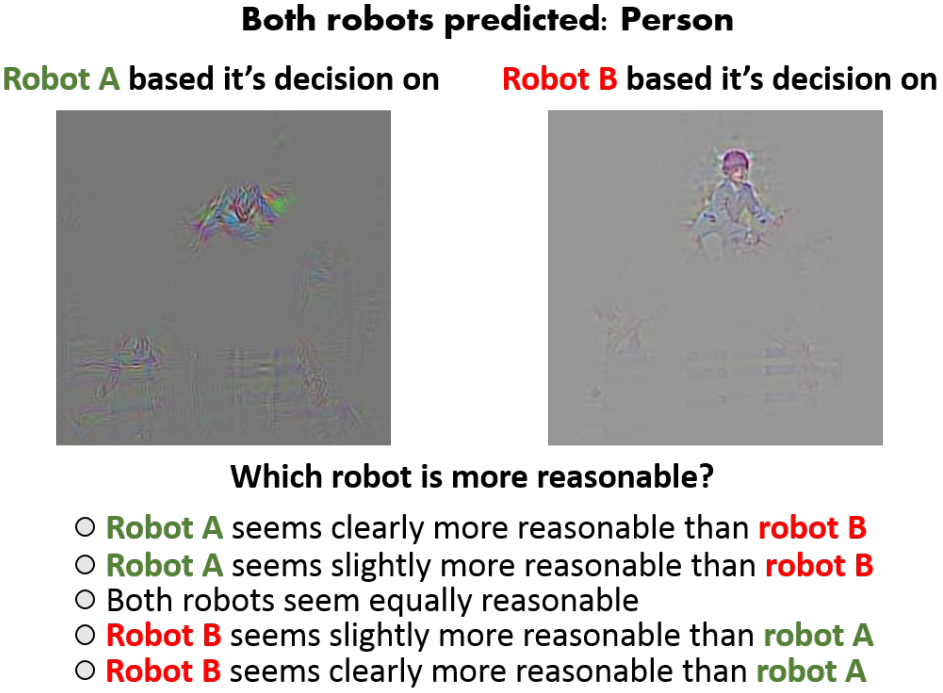
In this section, we describe the human studies and experiments we conducted to understand the interpretability *vs*. faithfulness tradeoff of our approach to model predictions. Our first human study evaluates the main premise of our approach – are Grad-CAM visualizations more class discriminative than previous techniques? Having established that, we turn to

[3]We use GoogLeNet finetuned on COCO, as provided by [63].

[4]c-MWP [63] highlights arbitrary regions for predicted but non-existent categories, unlike Grad-CAM maps which typically do not.

(a) Raw input image. Note that this is not a part of the tasks (b) and (c)

(b) AMT interface for evaluating the class-discriminative property



(c) AMT interface for evaluating if our visualizations instill trust in an end user

Figure 3.6: AMT interfaces for evaluating different visualizations for class discrimination (b) and trustworthiness (c). Guided Grad-CAM outperforms baseline approaches (Guided-backprop and Deconvolution) showing that our visualizations are more class-discriminative and help humans place trust in a more accurate classifier.

understanding whether it can lead an end user to trust the visualized models appropriately.

For these experiments, we compare VGG-16 and AlexNet finetuned on PASCAL VOC

2007 `train` and visualizations evaluated on `val`.

3.4.1 Evaluating Class Discrimination

In order to measure whether Grad-CAM helps distinguish between classes, we select images from the PASCAL VOC 2007 val set, which contain exactly 2 annotated categories and create visualizations for each one of them. For both VGG-16 and AlexNet CNNs, we obtain category-specific visualizations using four techniques: Deconvolution, Guided Backpropagation, and Grad-CAM versions of each of these methods (Deconvolution Grad-CAM and Guided Grad-CAM). We show these visualizations to 43 workers on Amazon Mechanical Turk (AMT) and ask them "Which of the two object categories is depicted in the image?" (shown in Fig. 3.6).

Intuitively, a good prediction explanation is one that produces discriminative visualizations for the class of interest. The experiment was conducted using all 4 visualizations for 90 image-category pairs (*i.e*. 360 visualizations); 9 ratings were collected for each image, evaluated against the ground truth and averaged to obtain the accuracy in Table. 3.2. When viewing Guided Grad-CAM, human subjects can correctly identify the category being visualized in 61.23% of cases (compared to 44.44% for Guided Backpropagation; thus, Grad-CAM improves human performance by 16.79%). Similarly, we also find that Grad-CAM helps make Deconvolution more class-discriminative (from $53.33\% \rightarrow 60.37\%$). Guided Grad-CAM performs the best among all methods. Interestingly, our results indicate that Deconvolution is more class-discriminative than Guided Backpropagation (53.33% *vs*. 44.44%), although Guided Backpropagation is more aesthetically pleasing. To the best of our knowledge, our evaluations are the first to quantify this subtle difference.

3.4.2 Evaluating Trust

Given two prediction explanations, we evaluate which seems more trustworthy. We use AlexNet and VGG-16 to compare Guided Backpropagation and Guided Grad-CAM visu-

Table 3.2: Quantitative Visualization Evaluation. Guided Grad-CAM enables humans to differentiate between visualizations of different classes (Human Classification Accuracy) and pick more reliable models (Relative Reliability). It also accurately reflects the behavior of the model (Rank Correlation w/ Occlusion).

Method	**Human Classification Accuracy**	**Relative Reliability**	**Rank Correlation w/ Occlusion**
Guided Backpropagation	44.44	+1.00	0.168
Guided Grad-CAM	61.23	+1.27	0.261

alizations, noting that VGG-16 is known to be more reliable than AlexNet with an accuracy of 79.09 mAP (*vs*. 69.20 mAP) on PASCAL classification. In order to tease apart the efficacy of the visualization from the accuracy of the model being visualized, we consider only those instances where *both* models made the same prediction as ground truth. Given a visualization from AlexNet and one from VGG-16, and the predicted object category, 54 AMT workers were instructed to rate the reliability of the models relative to each other on a scale of clearly more/less reliable (+/-2), slightly more/less reliable (+/-1), and equally reliable (0). This interface is shown in Fig. 3.6. To eliminate any biases, VGG-16 and AlexNet were assigned to be 'model-1' with approximately equal probability. Remarkably, as can be seen in Table. 3.2, we find that human subjects are able to identify the more accurate classifier (VGG-16 over AlexNet) *simply from the prediction explanations, despite both models making identical predictions.* With Guided Backpropagation, humans assign VGG-16 an average score of 1.00 which means that it is slightly more reliable than AlexNet, while Guided Grad-CAM achieves a higher score of 1.27 which is closer to saying that VGG-16 is clearly more reliable. Thus, our visualizations can help users place trust in a model that generalizes better, just based on individual prediction explanations.

3.4.3 Faithfulness *vs*. Interpretability

Faithfulness of a visualization to a model is its ability to accurately explain the function learned by the model. Naturally, there exists a trade-off between the interpretability and

faithfulness of a visualization – a more faithful visualization is typically less interpretable and viceversa. In fact, one could argue that a fully faithful explanation is the entire description of the model, which in the case of deep models is not interpretable/easy to visualize. We have verified in previous sections that our visualizations are reasonably interpretable. We now evaluate how faithful they are to the underlying model. One expectation is that our explanations should be locally accurate, *i.e*. in the vicinity of the input data point, our explanation should be faithful to the model [15].

For comparison, we need a reference explanation with high local-faithfulness. One obvious choice for such a visualization is image occlusion [10], where we measure the difference in CNN scores when patches of the input image are masked. Interestingly, patches which change the CNN score are also patches to which Grad-CAM and Guided Grad-CAM assign high intensity, achieving rank correlation 0.254 and 0.261 (*vs*. 0.168, 0.220 and 0.208 achieved by Guided Backpropagation, c-MWP and CAM respectively) averaged over 2510 images in the PASCAL 2007 val set. This shows that Grad-CAM is more faithful to the original model compared to prior methods. Through localization experiments and human studies, we see that Grad-CAM visualizations are *more interpretable*, and through correlation with occlusion maps, we see that Grad-CAM is *more faithful* to the model.

3.5 Diagnosing image classification CNNs with Grad-CAM

In this section we further demonstrate the use of Grad-CAM in analyzing failure modes of image classification CNNs, understanding the effect of adversarial noise, and identifying and removing biases in datasets, in the context of VGG-16 pretrained on imagenet.

3.5.1 Analyzing failure modes for VGG-16

In order to see what mistakes a network is making, we first get a list of examples that the network (VGG-16) fails to classify correctly. For these misclassified examples, we use Guided Grad-CAM to visualize both the correct and the predicted class. As seen in Fig. 3.7,

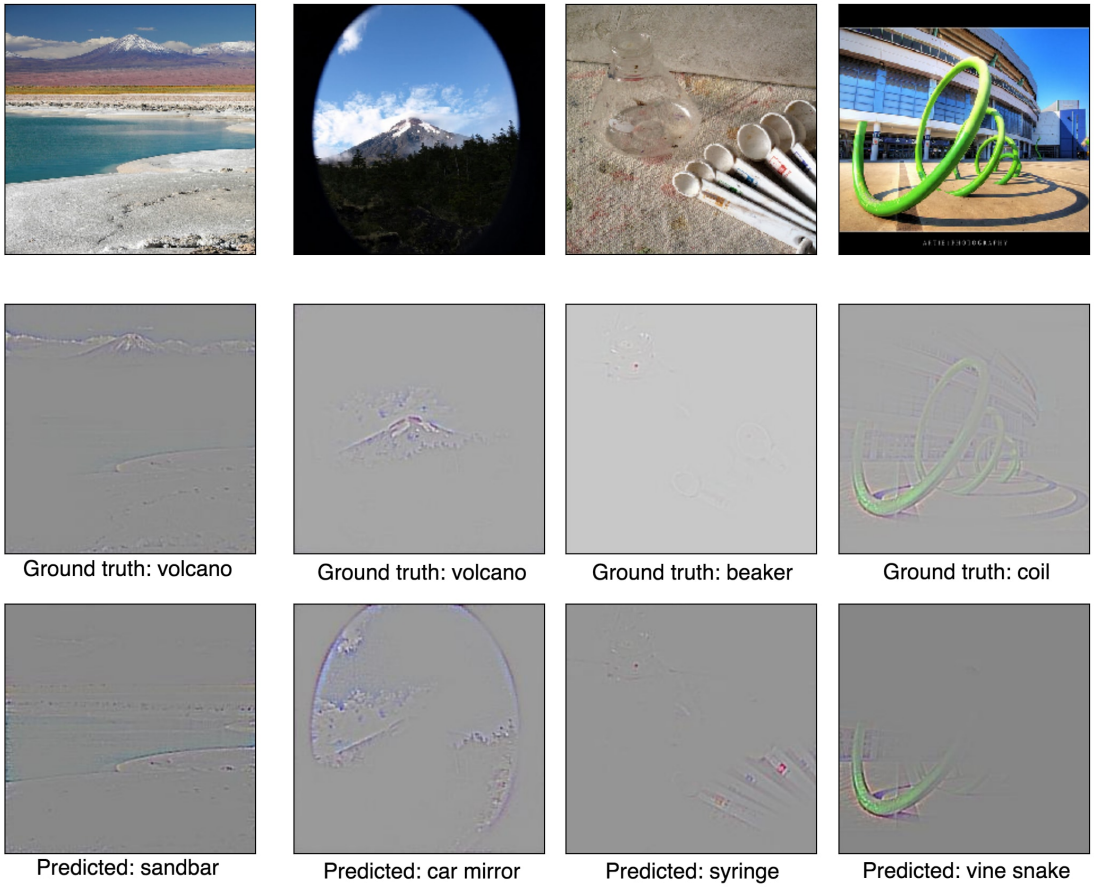


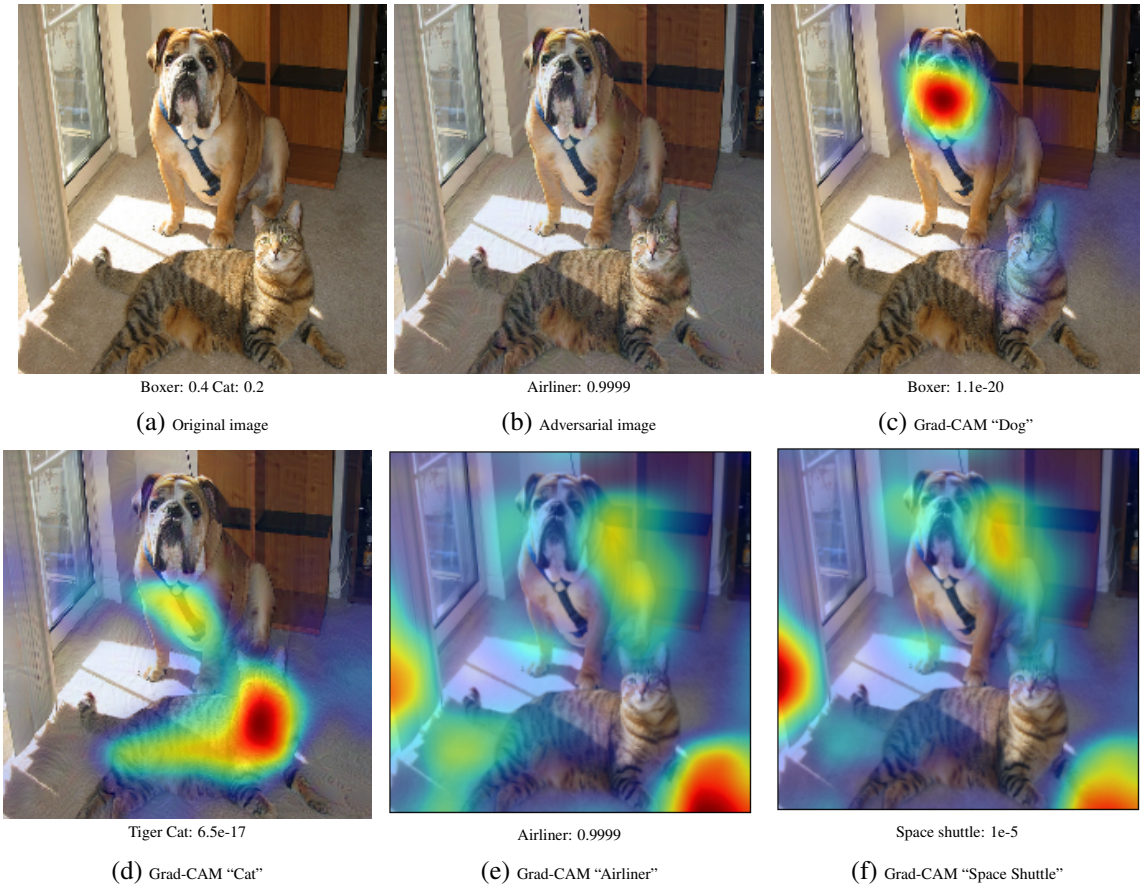
Figure 3.7: In these cases the model (VGG-16) failed to predict the correct class in its top 1 (a and d) and top 5 (b and c) predictions. Humans would find it hard to explain some of these predictions without looking at the visualization for the predicted class. But with Grad-CAM, these mistakes seem justifiable.

some failures are due to ambiguities inherent in ImageNet classification. We can also see that *seemingly unreasonable predictions have reasonable explanations*, an observation also made in HOGgles [66]. A major advantage of Guided Grad-CAM visualizations over other methods is that due to its high-resolution and ability to be class-discriminative, it readily enables these analyses.

3.5.2 Effect of adversarial noise on VGG-16

Goodfellow *et al*. [67] demonstrated the vulnerability of current deep networks to adversarial examples, which are slight imperceptible perturbations of input images that fool the

network into misclassifying them with high confidence. We generate adversarial images for an ImageNet-pretrained VGG-16 model such that it assigns high probability (> 0.9999) to a category that is not present in the image and low probabilities to categories that are present. We then compute Grad-CAM visualizations for the categories that are present. As shown in Fig. 3.8, despite the network being certain about the absence of these categories ('tiger cat' and 'boxer'), Grad-CAM visualizations can correctly localize them. This shows that Grad-CAM is fairly robust to adversarial noise.



(a) Original image (b) Adversarial image (c) Grad-CAM "Dog"

(d) Grad-CAM "Cat" (e) Grad-CAM "Airliner" (f) Grad-CAM "Space Shuttle"

Figure 3.8: (a-b) Original image and the generated adversarial image for category "airliner". (c-d) Grad-CAM visualizations for the original categories "tiger cat" and "boxer (dog)" along with their confidence. Despite the network being completely fooled into predicting the dominant category label of "airliner" with high confidence (> 0.9999), Grad-CAM can localize the original categories accurately. (e-f) Grad-CAM for the top-2 predicted classes "airliner" and "space shuttle" seems to highlight the background.

3.5.3 Identifying bias in dataset

In this section, we demonstrate another use of Grad-CAM: identifying and reducing bias in training datasets. Models trained on biased datasets may not generalize to real-world scenarios, or worse, may perpetuate biases and stereotypes (w.r.t. gender, race, age, *etc*.). We finetune an ImageNet-pretrained VGG-16 model for a "doctor" *vs*. "nurse" binary classification task. We built our training and validation splits using the top 250 relevant images (for each class) from a popular image search engine. And the test set was controlled to be balanced in its distribution of genders across the two classes. Although the trained model achieves good validation accuracy, it does not generalize well (82% test accuracy).

Grad-CAM visualizations of the model predictions (see the red box[5] regions in the middle column of Fig. 3.9) revealed that the model had learned to look at the person's face / hairstyle to distinguish nurses from doctors, thus learning a gender stereotype. Indeed, the model was misclassifying several female doctors to be a nurse and male nurses to be a doctor. Clearly, this is problematic. Turns out the image search results were gender-biased (78% of images for doctors were men, and 93% images for nurses were women).

Through these intuitions gained from Grad-CAM visualizations, we reduced bias in the training set by adding in images of male nurses and female doctors, while maintaining the same number of images per class as before. The re-trained model not only generalizes better (90% test accuracy), but also looks at the right regions (last column of Fig. 3.9). Additional analysis along with more Grad-CAM visualizations from both models can be found in [65]. This experiment demonstrates a proof-of-concept that Grad-CAM can help detect and remove biases in datasets, which is important not just for better generalization, but also for fair and ethical outcomes as more algorithmic decisions are made in society.

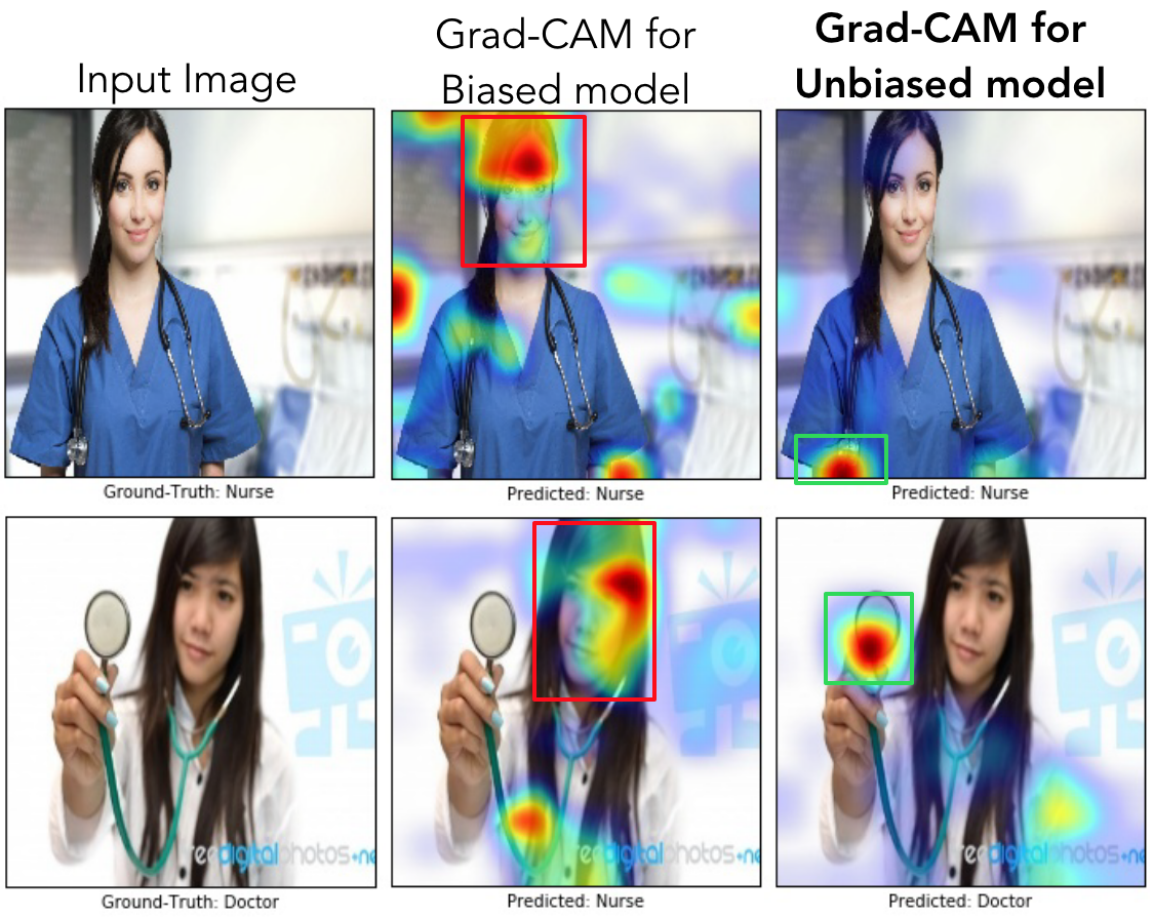


Figure 3.9: In the first row, we can see that even though both models made the right decision, the biased model (model1) was looking at the face of the person to decide if the person was a nurse, whereas the unbiased model was looking at the short sleeves to make the decision. For the example image in the second row, the biased model made the wrong prediction (misclassifying a doctor as a nurse) by looking at the face and the hairstyle, whereas the unbiased model made the right prediction looking at the white coat, and the stethoscope.

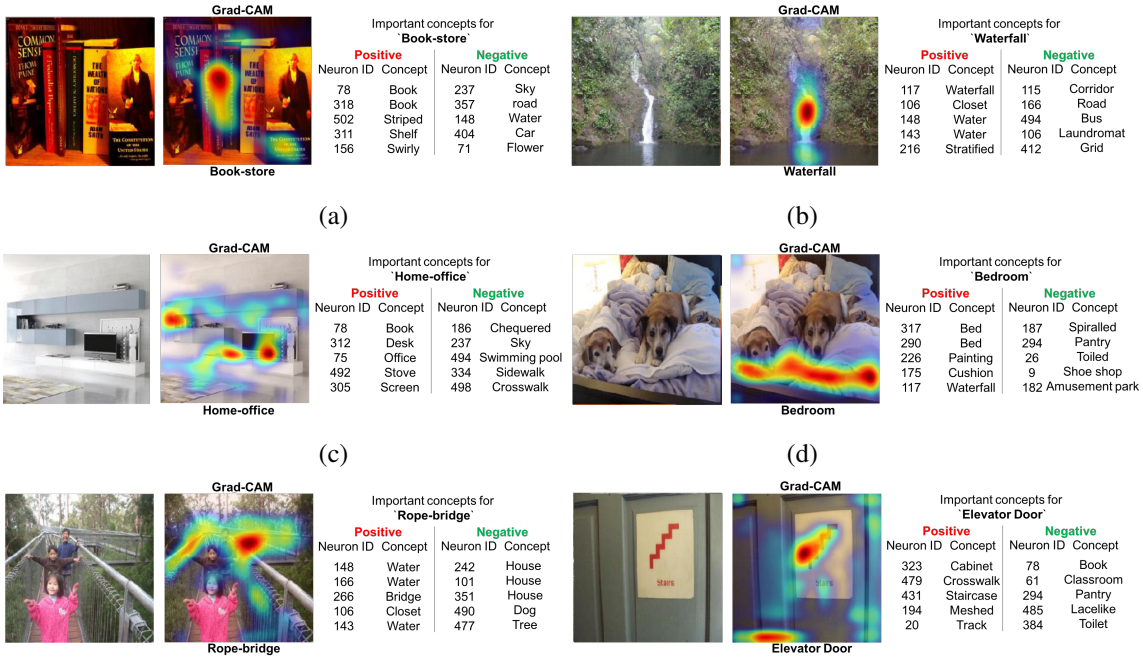


Figure 3.10: Examples showing visual explanations and textual explanations for VGG-16 trained on Places365 dataset [68]. For textual explanations we provide the most important neurons for the predicted class along with their names. Important neurons can be either be persuasive (positive importance) or inhibitive (negative importance). The first 2 rows show success cases, and the last row shows 2 failure cases. We see that in (a), the important neurons computed by (3.1) look for concepts such as book and shelf which are indicative of class 'Book-store' which is fairly intuitive.

3.6 Textual Explanations with Grad-CAM

Equation. (3.1) gives a way to obtain neuron-importance, α, for each neuron in a convolutional layer for a particular class. There have been hypotheses presented in the literature [69, 10] that neurons act as concept 'detectors'. Higher positive values of the neuron importance indicate that the presence of that concept leads to an increase in the class score, whereas higher negative values indicate that its absence leads to an increase in the score for the class.

Given this intuition, let's examine a way to generate textual explanations. In recent work, Bau *et al*. [58] proposed an approach to automatically name neurons in any convolutional layer of a trained network. These names indicate concepts that the neuron looks for

[5]The green and red boxes are drawn manually to highlight correct and incorrect focus of the model.

in an image. Using their approach. we first obtain neuron names for the last convolutional layer. Next, we sort and obtain the top-5 and bottom-5 neurons based on their class-specific importance scores, α_k. The names for these neurons can be used as text explanations.

Fig. 3.10 shows some examples of visual and textual explanations for the image classification model (VGG-16) trained on the Places365 dataset [68]. In (a), the positively important neurons computed by (3.1) look for intuitive concepts such as book and shelf that are indicative of the class 'Book-store'. Also note that the negatively important neurons look for concepts such as sky, road, water and car which don't occur in 'Book-store' images. In (b), for predicting 'waterfall', both visual and textual explanations highlight 'water' and 'stratified' which are descriptive of 'waterfall' images. (e) is a failure case due to misclassification as the network predicted 'rope-bridge' when there is no rope, but still the important concepts (water and bridge) are indicative of the predicted class. In (f), while Grad-CAM correctly looks at the door and the staircase on the paper to predict 'Elevator door', the neurons detecting doors did not pass the IoU threshold[6] of 0.05 (chosen in order to suppress the noise in the neuron names), and hence are not part of the textual explanations. More qualitative examples can be found in the Sec. A.7.

3.7 Grad-CAM for Image Captioning and VQA

Finally, we apply Grad-CAM to vision & language tasks such as image captioning [44, 46, 43] and Visual Question Answering (VQA) [1, 47, 48, 49]. We find that Grad-CAM leads to interpretable visual explanations for these tasks as compared to baseline visualizations which do not change noticeably across changing predictions. Note that existing visualization techniques either are not class-discriminative (Guided Backpropagation, Deconvolution), or simply cannot be used for these tasks/architectures, or both (CAM, c-MWP).

[6] Area of overlap between ground truth concept annotation and neuron activation over area of their union. More details of this metric can be found in [70]

(a) Image captioning explanations

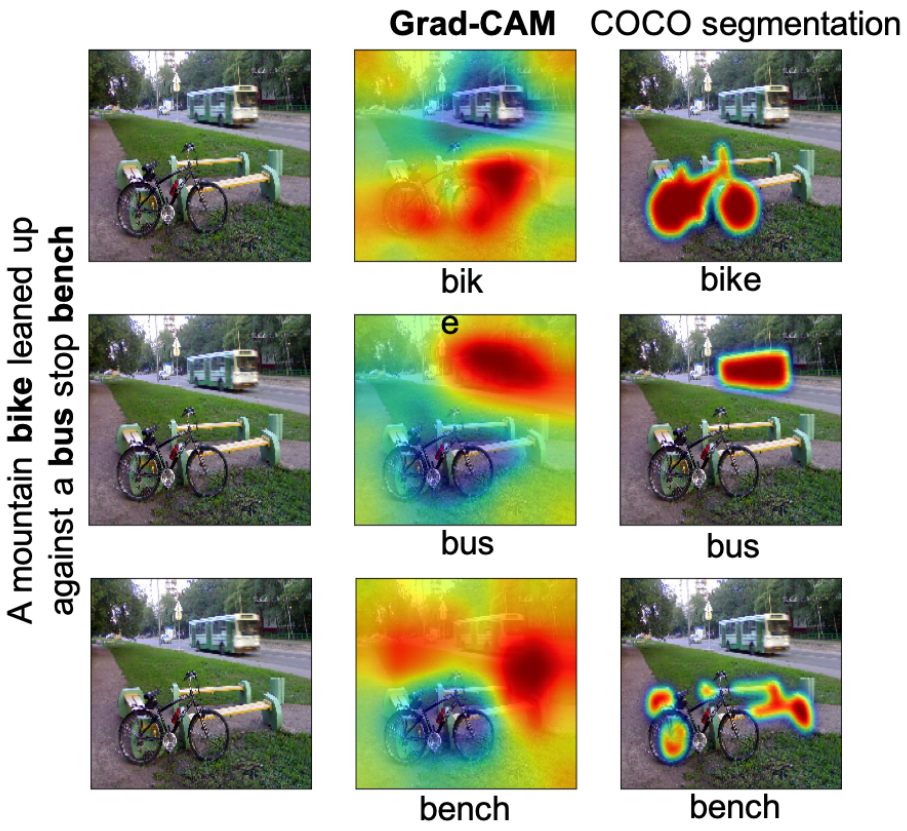
(b) Comparison to DenseCap

Figure 3.11: Interpreting image captioning models: We use our class-discriminative localization technique, Grad-CAM to find spatial support regions for captions in images. Fig. 3.11a Visual explanations from image captioning model [71] highlighting image regions considered to be important for producing the captions. Fig. 3.11b Grad-CAM localizations of a *global* or *holistic* captioning model for captions generated by a dense captioning model [46] for the three bounding box proposals marked on the left. We can see that we get back Grad-CAM localizations (right) that agree with those bounding boxes – even though the captioning model and Grad-CAM techniques do not use any bounding box annotations.

3.7.1 Image Captioning

In this section, we visualize spatial support for an image captioning model using Grad-CAM. We build Grad-CAM on top of the publicly available neuraltalk2[7] implementation [71] that uses a finetuned VGG-16 CNN for images and an LSTM-based language model. Note that this model does not have an explicit attention mechanism. Given a caption, we compute the gradient of its log probability w.r.t. units in the last convolutional layer of the CNN ($conv5\_3$ for VGG-16) and generate Grad-CAM visualizations as described in

[7] `https://github.com/karpathy/neuraltalk2`



(a)

Figure 3.12: Qualitative Results for our word-level captioning experiments: (a) Given the image on the left and the caption, we visualize Grad-CAM maps for the visual words "bike", "bench" and "bus". Note how well the Grad-CAM maps correlate with the COCO segmentation maps on the right column.

Section 6.6. See Fig. 3.11a. In the first example, Grad-CAM maps for the generated caption localize every occurrence of both the kites and people despite their relatively small size. In the next example, Grad-CAM correctly highlights the pizza and the man, but ignores the woman nearby, since 'woman' is not mentioned in the caption. More examples are in Sec. A.2.

Comparison to dense captioning

Johnson *et al*. [46] recently introduced the Dense Captioning (DenseCap) task that requires a system to jointly localize and caption salient regions in a given image. Their model

consists of a Fully Convolutional Localization Network (FCLN) that produces bounding boxes for regions of interest and an LSTM-based language model that generates associated captions, all in a single forward pass. Using DenseCap, we generate 5 region-specific captions per image with associated ground truth bounding boxes. Grad-CAM for a whole-image captioning model (neuraltalk2) should localize the bounding box the region-caption was generated for, which is shown in Fig. 3.11b. We quantify this by computing the ratio of mean activation inside *vs*. outside the box. Higher ratios are better because they indicate stronger attention to the region the caption was generated for. Uniformly highlighting the whole image results in a baseline ratio of 1.0 whereas Grad-CAM achieves 3.27 ± 0.18. Adding high-resolution detail gives an improved baseline of 2.32 ± 0.08 (Guided Backpropagation) and the best localization at 6.38 ± 0.99 (Guided Grad-CAM). Thus, Grad-CAM is able to localize regions in the image that the DenseCap model describes, even though the holistic captioning model was never trained with bounding-box annotations.

Grad-CAM for individual words of caption

In our experiment we use the Show and Tell model [43] pre-trained on MSCOCO without fine-tuning through the visual representation obtained from Inception [61] architecture. In order to obtain Grad-CAM map for individual words in the ground-truth caption we one-hot encode each of the visual words at the corresponding time-steps and compute the neuron importance score using Eq. (3.1) and combine with the convolution feature maps using Eq. (3.2).

Comparison to Human Attention We manually created an object category to word mapping that maps object categories like <person> to a list of potential fine-grained labels like [“child”, “man”, ”woman”, ...]. We map a total of 830 visual words existing in COCO captions to 80 COCO categories. We then use the segmentation annotations for the 80 categories as human attention for this subset of matching words.

We then use the pointing evaluation from [63]. For each visual word from the caption,

we generate the Grad-CAM map and then extract the maximally activated point. We then evaluate if the point lies within the human attention mapsegmentation for the corresponding COCO category, thereby counting it as a hit or a miss. The pointing accuracy is then calculated as

$Acc = \frac{\#Hits}{\#Hits+\#Misses}$. We perform this experiment on 1000 randomly sampled images from COCO dataset and obtain an accuracy of 30.0%. Some qualitative examples can be found in Fig. 3.12.

3.7.2 Visual Question Answering

Typical VQA pipelines [1, 47, 48, 49] consist of a CNN to process images and an RNN language model for questions. The image and the question representations are fused to predict the answer, typically with a 1000-way classification (1000 being the size of the answer space). Since this is a classification problem, we pick an answer (the score y^c in (3.3)) and use its score to compute Grad-CAM visualizations over the image to explain the answer. Despite the complexity of the task, involving both visual and textual components, the explanations (of the VQA model from Lu *et al*. [72]) described in Fig. 3.13 are surprisingly intuitive and informative. We quantify the performance of Grad-CAM via correlation with occlusion maps, as in Section 3.4.3. Grad-CAM achieves a rank correlation (with occlusion maps) of 0.60 ± 0.038 whereas Guided Backpropagation achieves 0.42 ± 0.038, indicating higher faithfulness of our Grad-CAM visualization.

We show qualitative examples comparing Guided Backpropagation with Grad-CAM and Guided Grad-CAM visualizations obtained for [72] in Fig. 3.14. Notice in the first row of Fig. 3.14, for the question, "*Is the person riding the waves?*", the VQA model with AlexNet and VGG-16 answered "No", as they concentrated on the person mainly, and not the waves. On the other hand, VGG-19 correctly answered "Yes", and it looked at the regions around the man in order to answer the question. In the second row, for the question, "*What is the person hitting?*", the VQA model trained with AlexNet answered

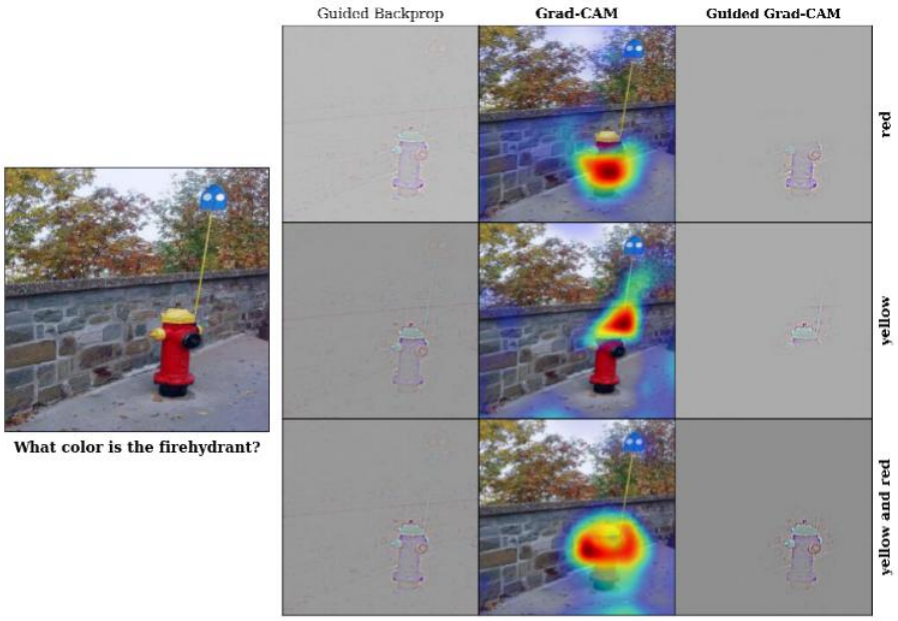
"Tennis ball" just based on context without looking at the ball. Such a model might be risky when employed in real-life scenarios. It is difficult to determine the trustworthiness of a model just based on the predicted answer. Our visualizations provide an accurate way to explain the model's predictions and help in determining which model to trust, without making any architectural changes or sacrificing accuracy. Notice in the last row of Fig. 3.14, for the question, "*Is this a whole orange?*", the model looks for regions around the orange to answer "No".

Comparison to Human Attention

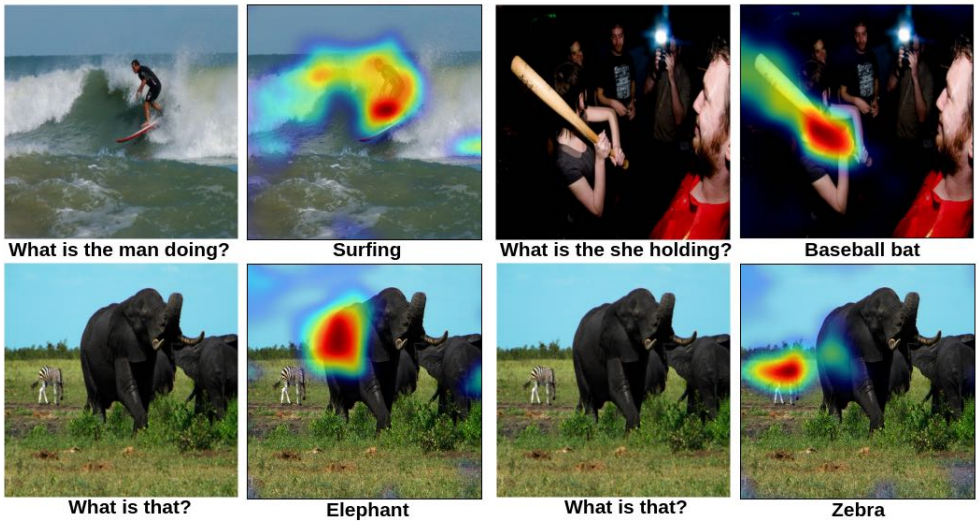
Das *et al*. [35] collected human attention maps for a subset of the VQA dataset [1]. These maps have high intensity where humans looked in the image in order to answer a visual question. Human attention maps are compared to Grad-CAM visualizations for the VQA model from [72] on 1374 val question-image (QI) pairs from [1] using the rank correlation evaluation protocol as in [35]. Grad-CAM and human attention maps have a correlation of 0.136, which is higher than chance or random attention maps (zero correlation). This shows that despite not being trained on grounded image-text pairs, even non-attention based CNN + LSTM based VQA models are surprisingly good at localizing regions for predicting a particular answer.

Visualizing ResNet-based VQA model with co-attention

Lu *et al*. [73] use a 200 layer ResNet [40] to encode the image, and jointly learn a hierarchical attention mechanism on the question and image. Fig. 3.13b shows Grad-CAM visualizations for this network. As we visualize deeper layers of the ResNet, we see small changes in Grad-CAM for most adjacent layers and larger changes between layers that involve dimensionality reduction. More visualizations for ResNets can be found in [65]. To the best of our knowledge, we are the first to visualize decisions from ResNet-based models.



(a) Visualizing VQA model from [72]



(b) Visualizing ResNet based Hierarchical co-attention VQA model from [73]

Figure 3.13: Qualitative Results for our VQA experiments: (a) Given the image on the left and the question "What color is the firehydrant?", we visualize Grad-CAMs and Guided Grad-CAMs for the answers "red", "yellow" and "yellow and red". Grad-CAM visualizations are highly interpretable and help explain any target prediction – for "red", the model focuses on the bottom red part of the firehydrant; when forced to answer "yellow", the model concentrates on it's top yellow cap, and when forced to answer "yellow and red", it looks at the whole firehydrant! (b) Our approach is capable of providing interpretable explanations even for complex models.

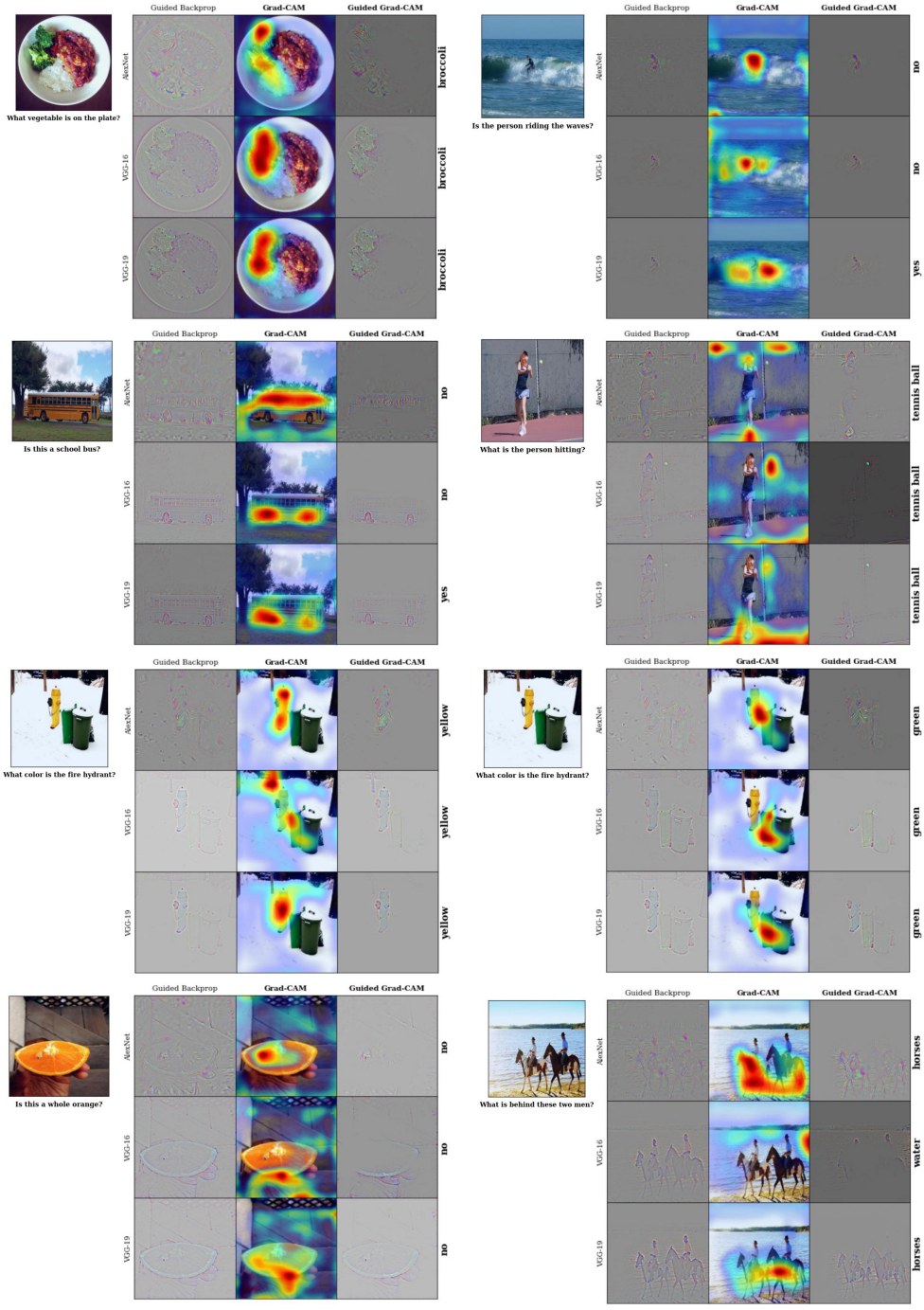


Figure 3.14: Guided Backpropagation, Grad-CAM and Guided Grad-CAM visualizations for the answers from a VQA model. For each image-question pair, we show visualizations for AlexNet, VGG-16 and VGG-19. Notice how the attention changes in row 3, as we change the answer from *Yellow* to *Green*.

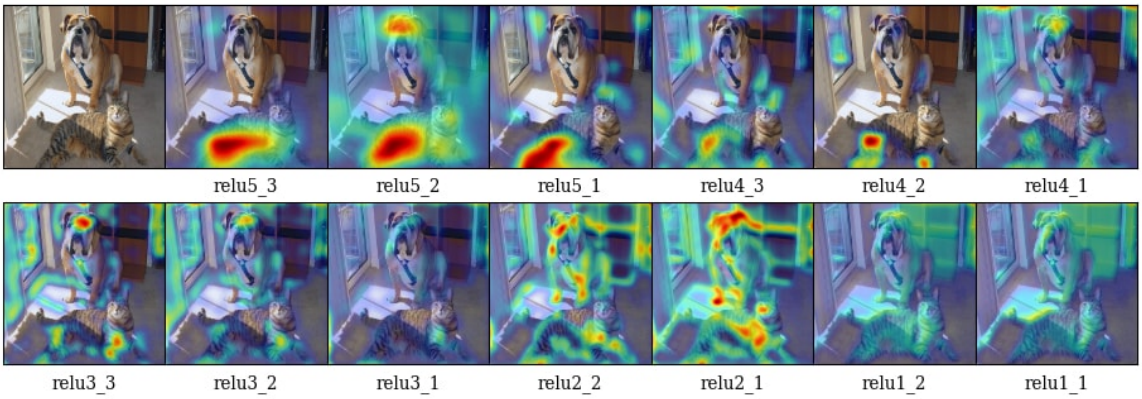


Figure 3.15: Grad-CAM at different convolutional layers for the 'tiger cat' class. This figure analyzes how localizations change qualitatively as we perform Grad-CAM with respect to different feature maps in a CNN (VGG16 [60]). We find that the best looking visualizations are often obtained after the deepest convolutional layer in the network, and localizations get progressively worse at shallower layers. This is consistent with our intuition that deeper convolutional layer capture more semantic concepts.

3.8 Ablation studies

We perform several ablation studies to explore and validate our design choices for computing Grad-CAM visualizations. This includes visualizing different layers in the network, understanding importance of ReLU in (3.2), analyzing different types of gradients (for ReLU backward pass), and different gradient pooling strategies.

1. Grad-CAM for different layers

We show Grad-CAM visualizations for the "tiger-cat" class at different convolutional layers in AlexNet and VGG-16. As expected, the results from Fig. 3.15 show that localization becomes progressively worse as we move to earlier convolutional layers. This is because later convolutional layers better capture high-level semantic information while retaining spatial information than earlier layers, that have smaller receptive fields and only focus on local features.

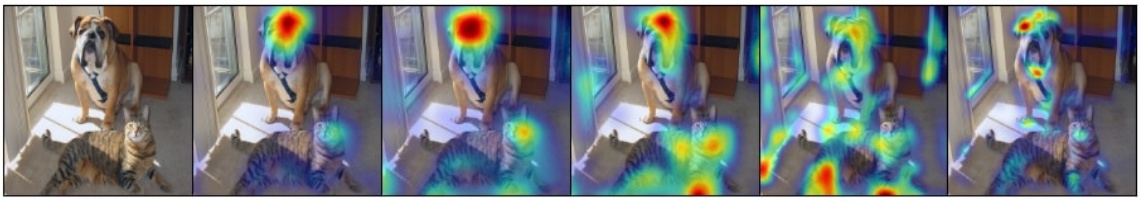


Figure 3.16: Grad-CAM localizations for "tiger cat" category for different rectified convolutional layer feature maps for AlexNet.

Table 3.3: Localization results on ILSVRC-15 val for the ablations. Note that this evaluation is over 10 crops, while visualizations are single crop.

Method	**Top-1 Loc error**
Grad-CAM	59.65
Grad-CAM without ReLU in Eq.1	74.98
Grad-CAM with Absolute gradients	58.19
Grad-CAM with GMP gradients	59.96
Grad-CAM with Deconv ReLU	83.95
Grad-CAM with Guided ReLU	59.14

2. Design choices

We evaluate different design choices via top-1 localization errors on the ILSVRC-15 val set [57]. See Table. 3.3.

2.1. Importance of ReLU in (3.3)

Removing ReLU ((3.3)) increases error by 15.3%. Negative values in Grad-CAM indicate confusion between multiple occurring classes.

2.2. Global Average Pooling vs. *Global Max Pooling*

Instead of Global Average Pooling (GAP) the incoming gradients to the convolutional layer, we tried Global Max Pooling (GMP). We observe that using GMP lowers the localization ability of Grad-CAM. An example can be found in Fig. 3.17 below. This may be due to the fact that max is statistically less robust to noise compared to the averaged gradient.

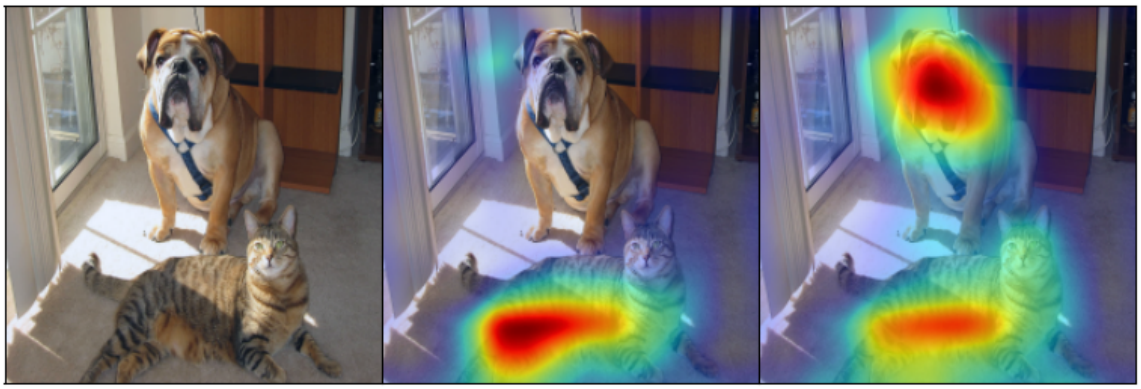


Figure 3.17: Grad-CAM visualizations for "tiger cat" category with Global Average Pooling and Global Max Pooling.

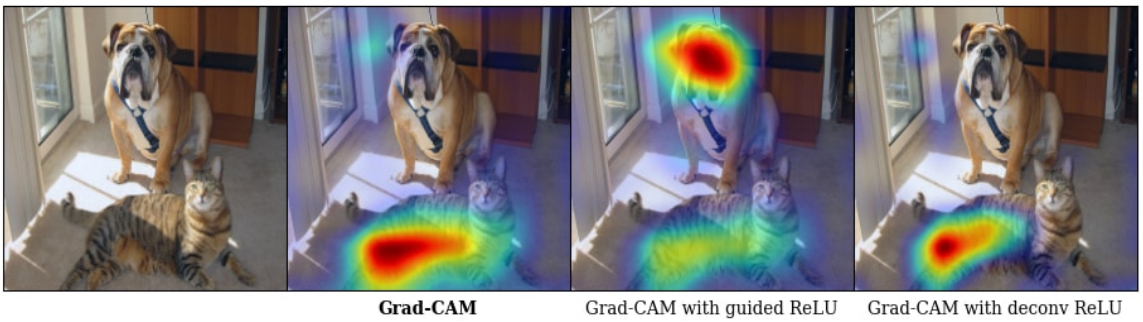


Figure 3.18: Grad-CAM visualizations for "tiger cat" category for different modifications to the ReLU backward pass. The best results are obtained when we use the actual gradients during the computation of Grad-CAM.

2.3. Effect of different ReLU on Grad-CAM

We experiment with Guided-ReLU [9] and Deconv-ReLU [10] as modifications to the backward pass of ReLU.

Guided-ReLU: Springenberg *et al*. [9] introduced Guided Backprop, where the backward pass of ReLU is modified to only pass positive gradients to regions of positive activations. Applying this change to the computation of Grad-CAM introduces a drop in the class-discriminative ability as can be seen in Fig. 3.18, but it marginally improves localization performance as can be seen in Table. 3.3.

Deconv-ReLU: In Deconvolution [10], Zeiler and Fergus introduced a modification to the backward pass of ReLU to only pass positive gradients. Applying this modification to the computation of Grad-CAM leads to worse results (Fig. 3.18). This indicates that

negative gradients also carry important information for class-discriminativeness.

3.9 Conclusion

In this work, we proposed a novel class-discriminative localization technique – Gradient-weighted Class Activation Mapping (Grad-CAM) – for making *any* CNN-based model more transparent by producing visual explanations. Further, we combined Grad-CAM localizations with existing high-resolution visualization techniques to obtain the best of both worlds – high-resolution and class-discriminative Guided Grad-CAM visualizations. Our visualizations outperform existing approaches on both axes – interpretability and faithfulness to original model. Extensive human studies reveal that our visualizations can discriminate between classes more accurately, better expose the trustworthiness of a classifier, and help identify biases in datasets. Further, we devise a way to identify important neurons through Grad-CAM and provide a way to obtain textual explanations for model decisions. Finally, we show the broad applicability of Grad-CAM to various off-the-shelf architectures for tasks such as image classification, image captioning and visual question answering. Finally, we provide several quantitative and qualitative results on interpreting predictions from off-the-shelf available image classification, image captioning and visual question answering models including visualizations from very deep architectures such as ResNets and their variants. Intriguingly, we find that our interpretations for visual question answering are better correlated to question-specific human attention maps than an attention-based VQA model [74]. We believe that a true AI system should not only be intelligent, but also be able to reason about its beliefs and actions for humans to trust and use it. Future work includes explaining decisions made by deep networks in domains such as reinforcement learning, natural language processing and video applications.

CHAPTER 4

FACILITATING KNOWLEDGE TRANSFER BETWEEN HUMANS AND AI

4.1 Introduction

Deep neural networks have pushed the boundaries of standard classification tasks in the past few years, with performance on many challenging benchmarks reaching near human-level accuracies. One caveat however is that these deep models require massive labeled datasets – failing to generalize from few examples or descriptions of unseen classes like humans can. To close this gap, the task of learning classifiers for unseen classes from external domain knowledge alone – termed zero-shot learning – has been the topic of increased interest within the community [75, 18, 19], [76], [77, 78], [79], [80], [81], [82], [83], [84, 85].

As humans, much of the way we acquire and transfer knowledge about novel concepts is in reference to or via composition of concepts which are already known. For instance, upon hearing that *"A Red Bellied Woodpecker is a small, round bird with a white breast, red crown, and spotted wings."*, we can compose our understanding of colors and birds to imagine how we might distinguish such an animal from other birds. However, applying a similar compositional learning strategy for deep neural networks has proven challenging.

While individual neurons in deep networks have been shown to learn localized, semantic concepts, these units lack referable groundings – *i.e*. even if a network contains units sensitive to *"white breast"* and *"red crown"*, there is no explicit mapping of these neurons to the relevant language name or description [58]. This observation encouraged prior work in interpretability to crowd-source "neuron names" to discover these groundings [58]. However, this annotation process needs to be re-executed for every trained model, which is expensive and impractical to learn unseen classes on the variety existing benchmarks

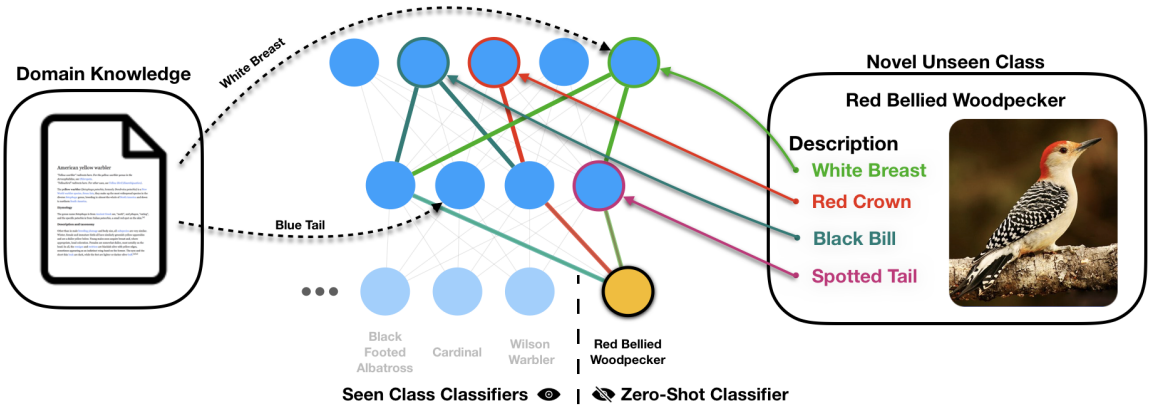


Figure 4.1: We present our Neuron Importance-aware Weight Transfer (NIWT) approach which maps free-form domain knowledge about unseen classes to relevant concept-sensitive neurons within a pretrained deep network. We then optimize the weights of a novel classifier such that the activation of this set of neurons results in high output scores for the unseen class. We present results on zero-short learning tasks, where no image instances of the unseen classes are used.

that have been proposed. Moreover, even if given perfect "neuron naming", it is an open question how to leverage this neuron-level descriptive supervision to train a classifier for an unseen class. This question is at the heart of our work.

Many existing zero-shot learning approaches make use of deep features (*i.e.* vectors of activations from some late layer in a network pretrained on some large-scale task) to learn joint embeddings with class descriptions [86, 22, 82, 84, 87, 88, 89, 90]. These higher-level features collapse many underlying concepts in the pursuit of class discrimination; consequentially, accessing lower-level concepts and recombining them in new ways to represent novel classes is difficult with these features. Mapping class descriptions to lower-level activations directly on the other hand is complicated by the high in-class variance of activations due to both spatial and visual differences within instances of a class. Apart from the non-linearity of the mapping at this low-level, this high in-class invariance also requires spatial attention in the mapping process. Our goal is to address these challenges by grounding class descriptions (including semantic attributes and free-form text) to the *importance* of lower-layer neurons to final class decisions [65].

In our approach, which we call Neural Importance-based Weight Transfer (NIWT), we learn a mapping between class-specific domain knowledge and the importances of individ-

ual neurons within a deep network. This mapping is learnt using images and corresponding domain knowledge of training classes and is expected to generalize to predict new classifiers with an arbitrary domain knowledge of an unseen class. We then use this learned mapping to predict neuron importances from knowledge about unseen classes and optimize classification weights such that the resulting network aligns with the predicted importances. In other words, based on domain-knowledge of the unseen categories, we can predict which low-level neurons should matter in the final classification decision. We can then learn classification weights such that the neurons predicted to matter do in fact matter. In this way, we connect the description of a previous unseen category to weights of a classifier that can predict this category at test time – all without having seen a single image from this category. To the best of our knowledge, this is the first zero-shot learning approach to align domain knowledge to intermediate neurons within a deep network. As an additional benefit, the learned mapping from domain knowledge to neuron importances grounds the neurons in interpretable semantics; automatically learning 'neuron names'.

We focus on the challenging generalized zero-shot (GZSL) learning setting. Unlike standard ZSL settings which evaluate performance only on unseen classes, GZSL considers both unseen and seen classes to measure the performance. In effect, GZSL is made more challenging by dropping the unrealistic assumption that test instances are known a priori to be from unseen classes in standard ZSL. We validate our approach across multiple standard datasets (CUBirds and AWA2) datasets and demonstrate superior performance to existing methods. Moreover, we evaluate the quality of neuron names as grounded explanations for classifier decisions through textual and visual examples.

Contributions. Concretely, we make the following contributions in this work:

- We introduce a zero-short learning approach based on mapping unseen class descriptions to neural importance within a deep network and then optimizing unseen classifier weights to effectively combine these concepts. In contrast to existing approaches, our

method is capable of explaining its zero-shot predictions with human-interpretable semantics from attributes or captions.

- We demonstrate the effectiveness of our approach by reporting state-of-the-art results on generalized zero-shot learning on CUB and AWA2 without altering the classifier weights for the 'seen' classes. We also show our approach can handle arbitrary forms of domain knowledge including attributes and image captions for unseen classes.

- We show how inverse mappings from neuron importance to domain knowledge can also be learned to provide interpretable explanations for the decisions made by newly learned classifiers for unseen classes.

4.2 Related Work

Attribute-based Zero-Shot Learning. One long-pursued way of recognizing object instances from previously unseen test categories (the zero-shot learning problem) is by leveraging knowledge about common attributes and shared parts (e.g., furry, striped, etc.). Earlier approaches (e.g., [18, 19, 20, 21]) model attributes as an intermediate layer that bridges the image features and class labels. More recently, several researchers have realized the limitation of the conditional independence assumption between image representation and class labels given the attributes [22, 23]. These methods usually model attributes in a continuous space with a core goal to learn a transformation matrix $\mathbf{W}$ mapping attributes to images. Other similar approaches utilized hyper-graph representations built on top of attributes and class labels (e.g., [91, 92]). Transformation-based approaches have recently shown a clearly better performances compared to graph based approaches as they are simpler and more efficient especially for fine-grained zero-shot recognition(e.g., [83, 82, 80, 86]).

Text-Based Zero-Shot Learning (ZSL). In a parallel research, pure text articles extracted from the web are leveraged instead of attributes to predict zero-shot visual classifiers [88]. In contrast to attributes based methods, text-based ZSL methods do not require or use any explicit attributes. The description of a new category is purely textual and could be extracted easily by just finding an web article about the class from the web (e.g., Wikipedia). A variety of approaches have been explored to study this task, Elhoseiny *et al*. [88] proposed an early method to that combines regression and domain transfer and predicts a classifier for a visual class given a TF-IDF textual representation of its corresponding Wikipedia article. More recent approaches adopted deep neural networks to learn convolutional classifiers, leading to a noticeable improvement on zero-shot accuracy (Bo *et al*. [93]). The proposed approaches are mainly based on learning a compatibility/similarity function between text descriptions and images either linearly [88] or non-linearly via deep neural networks [93] or kernels [90]. The classification is performed by associating the test image to the class that has the highest similarity to the corresponding class-level text / Wikipedia article.

Recently, Qiao *et al*. [87] visited the importance of sparsity regularization on zero-shot learning context. They demonstrated that noise emerging from non-visual terms in these Wikipedia Articles could be suppressed by promoting group sparsity in the connection between visual features and the text terms. Qiao *et al*.'s approach has actually started from an activation regularization proposed by [83] (applied in attribute-based ZSL) and added that additional group sparsity regularizer to improve the performance. Very recently, Elhoseiny *et al*. [89] observed that a similar noise suppression mechanism could be adopted to allow text-based ZSL at the part level but by encouraging this group sparsity at the level of every text_term-part pairs. This allows terms like "beak" to be connected to the head parts of the bird in a weakly supervised manner. Scott Reed *et al*. [94] have recently shown that by collecting 10 sentences per-image, their sentence-based zero-shot approach outperforms competitive to attribute-based methods zero-shot classification on the Caltech-

UCSD Birds.

In contrast to these approaches, we directly map text-based domain knowledge (captions) to internal components (neurons) of deep neural networks rather than learning associative mappings between images and text – offering not only comparable performance but also interpretability in our novel classifiers.

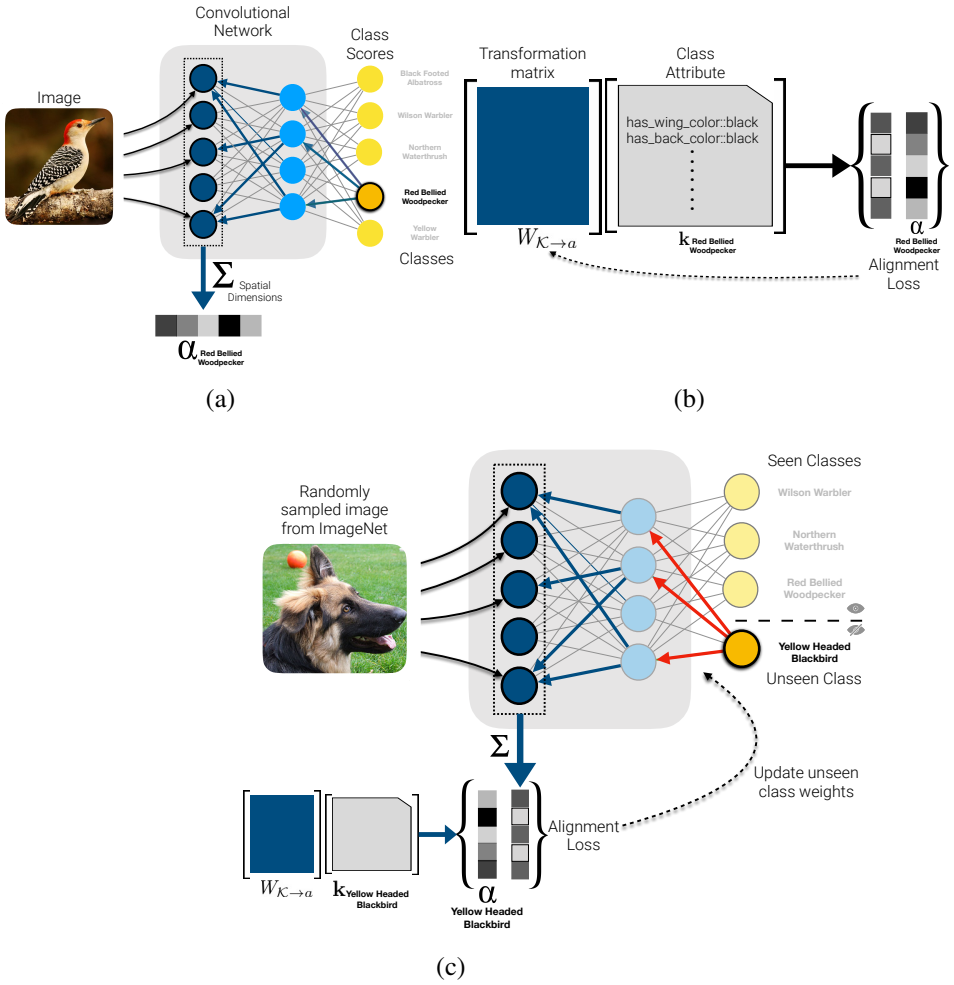


Figure 4.2: Our Neuron Importance-Aware Weight Transfer (NIWT) approach can be broken down in to three stages. a) class-specific neuron importances are extracted for seen classes at a fixed layer, b) a linear transform is learned to project free-form domain knowledge to these extracted importances, and c) weights for new classifiers are optimized such that neuron importances match those predicted by this mapping for unseen classes.

4.3 Neuron Importance-Aware Weight Transfer (NIWT)

In this section, we describe our Neuron Importance-Aware Weight Transfer (NIWT) approach to zero-shot learning. At a high level, NIWT maps free-form domain knowledge to neurons within a deep network and then learns classifiers based on novel class descriptions which respect these groundings. Concretely, NIWT consists of three steps:

1) estimating individual neuron importance to final network decisions for seen classes at a fixed network layer (see Figure 4.2a),

2) learning a linear mapping between domain knowledge and these importances (see Figure 4.2b), and

3) optimizing classifier weights with respect to predicted importances for unseen classes (see Figure 4.2c).

We discuss details of each in the following sections, but first recap the generalized zero-shot learning setting briefly and establish notation.

4.3.1 Preliminaries: Generalized Zero-Shot Learning (GZSL)

Consider a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ comprised of example input-output pairs from a set of *seen classes* $\mathcal{S} = \{1, \ldots, s\}$ and *unseen classes* $\mathcal{U} = \{s{+}1, \ldots, s{+}u\}$. For convenience, we use the subscripts $\mathcal{S}$ and $\mathcal{U}$ to indicate subsets corresponding to seen and unseen classes respectively, *e.g.* $\mathcal{D}_S = \{(x_i, y_i) \mid y_i \in \mathcal{S}\}$. Further, assume there exists domain knowledge $\mathcal{K} = \{k_1, ..., k_{s+u}\}$ corresponding to each class (*e.g.* class level attributes or natural language descriptions). Simply put, the goal of generalized zero-shot learning in this setting is to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{S} \cup \mathcal{U}$ from the input space $\mathcal{X}$ to the combined set of seen and unseen class labels using only the domain knowledge $\mathcal{K}$ and instances $\mathcal{D}_\mathcal{S}$ belonging to the seen classes.

4.3.2 Class-dependent Neuron Importance

Class descriptions like visual attributes or captions capture salient concepts about the content of corresponding images – for example, describing the coloration and shape of a bird's head. Similarly, a classifier must also learn some set of discriminative visual concepts in order to succeed; however, these concepts are not grounded in human interpretable language. In this stage, we identify neurons corresponding to these discriminative concepts before aligning them with domain knowledge in Section 4.3.3.

Consider a deep neural network $\texttt{NET}_{\mathcal{S}}(\cdot)$ trained for classification which predicts scores $\{o_c \mid c \in \mathcal{S}\}$ for the set of seen classes $\mathcal{S}$. One intuitive measure of a neuron n's importance to the final score o_c is simply the gradient of o_c with respect to the neuron's activation a^n. For networks containing convolutional units (which are replicated spatially), we follow [65] and simply compute importance as the mean gradient, writing the importance α_c^n as

$$\alpha_c^n = \overbrace{\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}}^{\text{global average pooling}} \underbrace{\frac{\partial o_c}{\partial a_{ij}^n}}_{\text{gradients via backprop}} \tag{4.1}$$

where $a_{i,j}^n$ is the activation of neuron n at spatial position i, j. For a given input, the importance of every neuron in the network can be computed for a given class via a single backward pass followed by a global average pooling operation for convolutional units. In practice, we focus on α's from single layers in the network in our experiments. We note that other measures of neuron importance have been proposed [95, 96] in various contexts; however, this simple gradient-based importance measure has some notable properties which we leverage in our approach.

Firstly, we find gradient-based importance scores to be quite consistent across images of the same class despite the visual variation between instances, and likewise to correlate poorly across classes. To assess this property quantitatively, we computed α's for neurons

in the final convolutional layer of a convolutional neural network trained on a fine-grained multi-class task (`conv5-3` of VGG-16 [97] trained on AWA2 [86]) for 10,000 randomly selected input images. We observed an average rank correlation of 0.817 for instances within the same class and 0.076 across classes. This relative invariance of α's to intra-class input variation may be due in part to the piece-wise linear decision boundaries learned in networks using ReLU [98] activations. As shown in [99], transitions between these linear regions are much less frequent between same-class inputs than across classes. Within the same linear region, activation gradients (and hence α's) are trivially identical.

Secondly, this measure of neural importance is fully differentiable with respect to model parameters which we leverage when learning novel classifiers (see Section 4.3.4).

4.3.3 Mapping Domain Knowledge to Neurons

As before, consider a deep neural network trained to classify instances of seen classes $\mathcal{S}$ and without loss of generality consider a single layer L within $\texttt{NET}_{\mathcal{S}}(\cdot)$. Given an instance $(x_i, y_i) \in \mathcal{D}_{\mathcal{S}}$, let $\mathbf{a}_c = \{\alpha_c^n \mid n \in L\}$ be a vector of importances computed for neurons in L with respect to class c when x_i is passed through the network. In this section, we learn simple linear mappings between domain knowledge and these importance vectors – aligning interpretable semantics with individual neurons.

To learn this mapping, we first compute the importance vector $\mathbf{a}_{y_i}$ for each seen class instance (x_i, y_i) and match it with the domain knowledge representation k_{y_i} of the corresponding class. Given this dataset of $(\mathbf{a}_{y_i}, k_{y_i})$ pairs, we learn a simple linear transform $W_{\mathcal{K}\to a}$ which map domain knowledge to importances. As importances are gradient based, we penalize errors in the predicted importances based on cosine distance, emphasizing alignment rather than matching exact values. We optimize the cosine distance loss between domain knowledge and importance vectors, *i.e.*

$$\mathcal{L}(\mathbf{a}_{y_i}, \mathbf{k}_{y_i}) = 1 - \frac{(W_{\mathcal{K}\to a} \cdot \mathbf{k}_{y_i}) \cdot \mathbf{a}_{y_i}}{\|W_{\mathcal{K}\to a} \cdot \mathbf{k}_{y_i}\| \, \|\mathbf{a}_{y_i}\|}, \tag{4.2}$$

via gradient descent to train $W_{\mathcal{K}\to a}$. We stop training when average rank-correlation of predicted and true importance vectors stabilize for a set of held out classes from $\mathcal{S}$.

Notably, this is a many-to-one mapping with the domain knowledge of one class needing to predict many different importance vectors. Despite this, this mapping achieves average rank correlations between true and predicted importance vectors of 0.2 to 0.5 for validation class instances. We explore the impact of error in importance vector prediction on weight optimization in Section 4.3.4. We also note that this simple linear mapping can be inverted to map neuron importances back to semantic concepts from the domain knowledge which we explore in Section 4.6.

4.3.4 Neural Importance to Classifier Weights

In this section, we use predicted importances to learn classifiers for the unseen classes. As these new classifiers will be built atop the trained seen-class network $\texttt{NET}_{\mathcal{S}}$, we modify $\texttt{NET}_{\mathcal{S}}$ to extend the output space to include the unseen class – expanding the final fully-connected layer to include additional neurons with weight vectors $\mathbf{w}^1, \ldots, \mathbf{w}^u$ for the unseen classes such that the network now additionally outputs scores $\{o_c \mid c \in \mathcal{U}\}$. We refer to this expanded network as $\texttt{NET}_{\mathcal{S}\cup\mathcal{U}}$. At this stage, the weights for the unseen classes are sampled randomly from a multivariate normal distribution with parameters estimated from the seen class weights and as such the output scores are uncalibrated and uninformative.

Given the learned mapping $W_{\mathcal{K}\to A}$ and unseen class domain knowledge $\mathcal{K}_{\mathcal{U}}$, we can predict unseen class importances $A_{\mathcal{U}} = \{\mathbf{a}_1, ..., \mathbf{a}_u\}$ with the importance vector for unseen class c predicted as $\mathbf{a}_c = W_{\mathcal{K}\to a}\mathbf{k}_c$. For a given input, we can compute importance vectors $\hat{a}_c$ for each unseen class c. As $\hat{\mathbf{a}}^c$ is a function of the weight parameters $\mathbf{w}_c$, we can simply supervise $\hat{\mathbf{a}}_c$ with the predicted importances $\mathbf{a}_c$ and optimize w^c with gradient descent – minimizing the cosine distance loss between predicted and observed importance vectors. However, the cosine distance loss does not account for scale and without regularization the scale of weights (and as consequence the outputs) of seen and unseen classes might vary

drastically, resulting in bias towards one set or the other.

To address this problem, we introduce a L_2 regularization term which constrains the learned unseen weights to be a similar scale as the mean of seen weights $\overline{\mathbf{w}}_{\mathcal{S}}$. We write the final objective as

$$\mathcal{L}(\hat{\mathbf{a}}_c, \mathbf{a}_c) = 1 - \frac{\hat{\mathbf{a}}_c \cdot \mathbf{a}_c}{\|\hat{\mathbf{a}}_c\| \, \|\mathbf{a}_c\|} + \lambda \|\mathbf{w}_c - \overline{\mathbf{w}}_{\mathcal{S}}\|, \tag{4.3}$$

where λ is controls the strength of this regularization. We examine the effect of this trade-off in Section 4.5.1, finding training to be robust to a wide range of λ values. We note that as observed importances $\mathbf{a}^c$ are themselves computed from network gradients, updating weights based on this loss requires computing a Hessian-vector product; however, this is relatively efficient as the number of weights for each unseen class is relatively small and independent with respect to those of other classes.

Training Images.

Note that to perform the optimization described above, we need to pass an image through the network to compute importance vectors. As noted in Section 4.3.2, importances are only weakly correlated with image features – as such, we find simply inputing images with natural statistics are sufficient. Specifically, we pair random images from ImageNet[57] with random tuples $(\hat{\mathbf{a}}_c, \mathbf{k}_c)$ to construct a dataset upon which we perform the importance to weight optimization.

4.4 Experiments

In this section, we evaluate our approach on generalized zero-shot learning (Section 4.4.1) and present analysis of each stage of our method (Section 4.5).

Algorithm 1 Neural Importance-aware Weight Transfer

Input: $\{I_S, y^S, \mathcal{K}_S\}$, $\{\mathcal{K}_U, \alpha_U^k\}$
Output: $\{y^U\}$

procedure NIWT ($\mathcal{K}, \alpha_U^k, W_f$)
Finetune network $(h_S(.) : I_S \rightarrow y^S)$
Obtain neuron-importance $\alpha_S^k \leftarrow \frac{1}{Z}\sum_i \sum_j \frac{\partial y^S}{\partial A_{conv5}^{i,j}}$
Learn $\mathbf{W}_k^\alpha : \mathcal{K}_S \rightarrow \alpha_S^k$
Obtain $\hat{\alpha}_U^k \leftarrow \mathbf{W}_k^\alpha(\mathcal{K}_U)$
Extend to new classes: $h_{S+U}(.) := h_S(.) \cup W_f^U$
Initialize $W_f^U \leftarrow \mathcal{N}(0, 1)$
$\alpha_U^k \leftarrow \frac{1}{Z}\sum_i \sum_j \frac{\partial y^U}{\partial A_{conv5}^{i,j}}$
    while $|\alpha_U^k - \hat{\alpha}_U^k| > \delta$ **do**
        Loss $L = l(\alpha_U^k, \hat{\alpha}_U^k)$
        $W_f^U \leftarrow W_f^U - \lambda \frac{dL}{dW_f^U}$
        $\alpha_U^k \leftarrow \frac{1}{Z}\sum_i \sum_j \frac{\partial y^U}{\partial A_{conv5}^{i,j}}$
    end
Return trained network: $h_{S+U}(.)$

4.4.1 Experimental Setting

Datasets and Metrics.

We conduct our GZSL experiments on the

- **Caltech-UCSD Birds 200 (CUB) [100]** – The CUB dataset consists of 11788 images corresponding to 200 species of birds. Each image has been annotated with 312 binary attribute labels which describe fine grained physical bird features such as the color and shape of specific body parts. Additionally, each image is associated with 10 human captions [94]. We evaluate our approach using both attributes and captions.
- **Animals with Attributes 2 (AWA2) [86]** – The AWA2 dataset consists of 37, 322 images of 50 animal species (on average 764 per class but with a wide range). Each class is labeled with 85 binary and continuous attributes.

For both datasets, we use the GZSL splits proposed in [86] which ensure that no unseen class occurs within the ImageNet [57] dataset which is commonly used for training classi-

fication networks for feature extraction. As in [79], we evaluate our approach using class-normalized accuracy computed over both seen and unseen classes (*i.e*. 200-way classification for CUB) – breaking the results down into unseen accuracy $\texttt{Acc}_{\mathcal{U}}$, seen accuracy $\texttt{Acc}_{\mathcal{S}}$, and the harmonic mean between them $\texttt{H}$.

Models.

We experiment with ResNet101 [40] and VGG16 [60] models pretrained on ImageNet [57] and fine-tuned on the seen classes. For each, we train a version by finetuning all layers and another by updating only the final classification weights. For VGG, both the finetuned and fixed achieve similar accuracies (74.84% finetuned vs 66.8% fixed for CUB and 92.32% vs 91.44% for AWA2). ResNet on the other hand sees sharp declines for fixed models (60.6% finetuned vs 28.26% fixed for CUB and 90.10% vs 70.7% for AWA2). We include more training details in the supplementary.

NIWT Settings.

To train the domain knowledge to importance mapping we hold out five seen classes and stop optimization when rank correlation between observed and predicted importances is highest. For attribute vectors, we use the class level attributes directly and for captions on CUB we use average word2vec embeddings[101] for each class. When optimizing for weights given importances, we stop when the loss fails to improve by 1% over 40 iterations. For a fixed learning rate ($1e^{-4}$), we vary the regularization coefficient λ from $1e^{-7}$ to $1e^{-2}$ and select the model with the lowest loss.

Baselines.

We compare NIWT with a number of well-performing zero-shot learning approaches based on learning joint embeddings of image features and class information. ALE [80], SJE [82], and DEVISE [81] all learn compatibility function for class labels and visual features using

Table 4.1: Generalized Zero-Shot Learning performances on the proposed splits [86] for CUB and AWA2. We report class-normalized accuracies on seen and unseen classes and harmonic mean.[1] reproduced from [86]. [2] based on code provided by the authors.

			CUB [100]			AWA2 [86]		
		Method	$\texttt{Acc}_{\mathcal{U}}$	$\texttt{Acc}_{\mathcal{S}}$	H	$\texttt{Acc}_{\mathcal{U}}$	$\texttt{Acc}_{\mathcal{S}}$	H
ResNet101 [40]	Fixed	ALE [80][1]	23.7	62.8	34.4	14.0	81.8	23.9
		SJE [82][1]	23.5	59.2	33.6	8.0	73.9	14.4
		DEVISE [81][1]	23.8	53.0	32.8	17.1	74.7	27.8
		Deep Embed. [102][2]	-	-	-	25.7	81.72	39.11
		NIWT-Attributes	14.26	10.37	12.01	24.30	46.38	31.9
	FT	Deep Embed. [102][2]	-	-	-	22.5	74.59	34.57
		NIWT-Attributes	10.22	56.16	17.3	14.73	54.69	23.21
		NIWT-Caption	16.9	40.00	23.7		N/A	
VGG16 [60]	Fixed	Deep Embed. [102][2]	-	-	-	28.99	41.65	34.18
		NIWT-Attributes	35.4	22.2	27.2	**47.55**	12.46	19.74
	FT	Deep Embed. [102][2]	-	-	-	28.6	42.4	34.19
		NIWT-Attributes	**37.3**	35.24	**36.23**	34.76	**83.21**	**49.03**
		NIWT-Caption	20.32	**38.9**	26.7		N/A	

some form of ranking loss. We take the results for these methods on the proposed split directly from [86].

We also compare against the recent Deep Embedding approach of [102] which also leverages deep networks, jointly aligning domain knowledge with deep features end-to-end. For this approach, we use code provided to us by the authors and perform a hyperparameter search for the proposed split as the original paper did not report on it. We were not able to find a configuration that achieved better than random for the proposed CUB split but we do not report this in Table 4.1.

4.4.2 Results

We show results in Table 4.1 for CUB and AWA2 using all model settings. There are a number of interesting trends to observe:

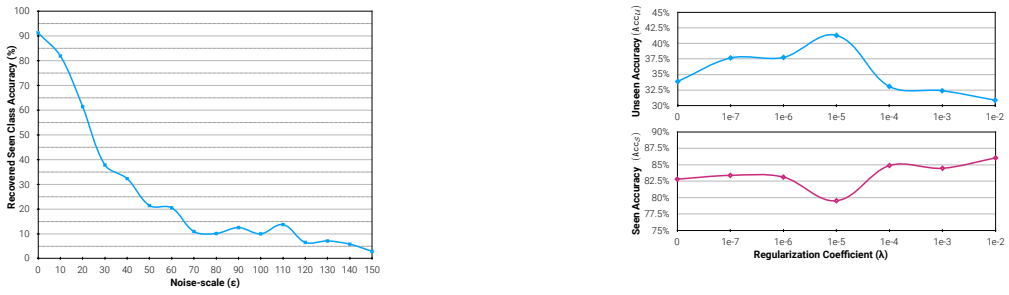
1. **NIWT sets the state of the art in generalized zero-shot learning.** For both datasets, NIWT-Attributes based on VGG establishes a new state of the art for harmonic mean (36.23% for CUB and 49.03% for AWA2). Moreover, the gap between NIWT and the next highest scoring method in AWA2 is quite large (approximately 15%).
2. **Finetuning generally improves performance - especially for VGG.** For CUB and AWA2, finetuning the VGG network offers significant gains in performance for NIWT (27.2% to 36.23% H on CUB and 19.74% to 49.03% H on AWA2); however, finetuning ResNet has mixed results with performance on AWA2 dropping somewhat (31.9% to 23.21% H) while performance on CUB rises sharply (12.01% to 23.7 %H).
3. **NIWT better leverages finetuned networks.** NIWT consistently makes higher gains in performance when switching to fully-finetuned networks compared to the Deep Embedding [102] method which either maintains or reduce performance compared to fixed networks.
4. **NIWT effectively grounds both attributes and free-form language.** We see strong performance both for attributes and captions across both networks (36.23% and 26.7% H for VGG and 17.3% and 23.7% H for ResNet respectively).

4.5 Analysis

To better understand the different stages of our approach, we perform a series of experiments to analyze and isolate individual components in our approach.

4.5.1 Effect of Regularization Coefficient λ.

One key component to our importance to weight optimization is the regularizer which enforces that learned unseen weights be close to the mean seen weight – avoiding arbitrary scaling of the learned weights and the bias this could introduce. To explore the effect of the regularizer, we vary the coefficient λ from 0 to $1e^{-2}$. Figure 4.3b shows the final seen and unseen class-normalized accuracy for the AWA2 dataset at convergence for different λ's.



(a) Noise Tolerance (ϵ)

(b) Regularizer Sensitivity (λ)

Figure 4.3: Analysis of the importance vector to weight optimization. (left) We find that ground-truth weights can be recovered for a pre-trained network even in the face of high noise. (right) We also show the importance of the regularization term to final model performance.

Without regularization ($\lambda = 0$) the unseen weights tend to be a bit too small and achieve an unseen accuracy of only 33.86%. As λ is increased the unseen accuracy grows until peaking at $\lambda = 1e^{-5}$ with an unseen accuracy of 41.28% – an improvement of over 8% from the unregularized version! Of course, this improvement comes with a trade-off in seen accuracy of about 3% over the same interval. As λ grows larger $> 1e^{-4}$, the regularization constraint becomes too strong and the optimization has trouble learning anything for the scene classes at all.

4.5.2 Noise Tolerance in Neuron Importance to weight optimization

One important component of NIWT is the ability to ground concepts learnt by a convolutional network in some referable domain. Due to the inherent noise involved in the $W_{\mathcal{K}\rightarrow A}$, the classifier obtained on unseen classes in the expanded network $\texttt{NET}_{\mathcal{S}\cup\mathcal{U}}$ is not entirely perfect. In order to judge the capacity of the optimization procedure, we experiment with a toy setting where we initialize an unseen classifier head with the same dimensionality as the seen classes and try to explicitly recover the seen class weights with supervision only from the *oracle* $\mathbf{a}_c$ obtained from the seen classifier head from the seen classes. To account for the error involved in estimating $\mathbf{a}_c$, we incorporate increasing levels of zero-centered gaussian noise in the same and study recovery performance in terms of accuracy of the recovered classifier head on the seen-test split. That is, the supervision from importance

vectors is constructed as follows:

$$\tilde{\mathbf{a}}_c = \mathbf{a}_c + \epsilon \overline{||\mathbf{a}_c||}_1 \mathcal{N}(0, I) \tag{4.4}$$

We operate at different values of ϵ, characterizing different levels of corruption of the supervision from $\mathbf{a}_c$ and observe recovery performance in terms of accuracy of the recovered classifier head. Fig. 4.3a shows the effect of noise on the ability to recover seen classifier weights (`fc7`) for a VGG-16 network trained on 40 seen classes of AWA2 dataset with the same objective as the one used for unseen classes.

In the absence of noise over $\mathbf{a}_c$ supervision, we find that we are exactly able to recover the seen class weights and are able to preserve the pre-trained accuracy on seen classes ($\sim 92.1\%$). If we increase the noise-level by a factor of 10 (adding noise to each dimension on the scale of 10% of a_c's average norm), we observe only minor reduction in the accuracy of the recovered seen class weights. As expected, this downward trend continues as we increase the noise-level until we reach almost chance-level performance on the recovered classifier head. This experiment shows that the importance vector to weights optimization is quite robust even to fairly extreme noise.

4.5.3 Network Depth of Importance Extraction.

In this section, we explore the sensitivity of NIWT with respect to the layer from which we extract importance vectors in the convolutional network. As an experiment (in addition to Table 4.1) we observe generalized zero-shot learning performance upon convergence on AWA2 by extracting importance vectors for classes from VGG-16 at different layers in the network. We observe that out of the ones we experimented with `conv5_3` performs the best with $\mathtt{H} = 49.03$ followed by `conv4_3` ($\mathtt{H} = 44.2$), `conv3_3` ($\mathtt{H} = 37.2$) and `conv2_2` ($\mathtt{H} = 28.1$). We also experimented with the layers `fc6` and `fc7` resulting in values of `H` being 28.6 and 0 respectively.

Note that performing NIWT on importance vectors extracted from the penultimate layer `fc7` is equivalent to learning the unseen head classifier weights directly from the domain space representation ($\mathbf{k}_c$). Consistent with our hypothesis, this performs very poorly across all the metrics with almost no learning involved for the unseen classes at all. We hypothesize that this is due to the restricted capacity of the linear transformation $W_{\mathcal{K}\to A}$ involved in the process.

4.5.4 Alpha to Weight Input Images

We evaluate performance with differing input images during weight optimization (random noise images, ImageNet images, and seen class images). We show performance of each in Table 4.2. As expected, performance improves as input images more closely resemble the unseen classes; however, we note that learning occurs even with random images.

Table 4.2: Results by sampling images on different sets for NIWT-Attributes on VGG-CUB.

Method	$\texttt{Acc}_{\mathcal{U}}$	$\texttt{Acc}_{\mathcal{S}}$	H
Random Normal	25.0	42.4	31.4
ImageNet	37.3	35.2	36.2
Seen-Classes	36.0	38.4	35.1

4.5.5 Behavior of NIWT across Iterations

To understand the behavior of NIWT we observe the variation of the seen and unseen class normalized accuracies over the iterations of the importance to weight optimization process. In Fig. 4.4, we plot $\texttt{Acc}_{\mathcal{U}}$, $\texttt{Acc}_{\mathcal{S}}$ and `H` after every 250 iterations with VGG16 as the base architecture on the AWA2 dataset.

We observe that there is a sharp rise in unseen class accuracy ($\sim 27\%$ at around 0.25k iterations) accompanied by a comparatively small drop in seen class accuracy ($\sim 1\%$ at around 0.25k iterations). However, as the iterations progress, the rise in unseen class accuracy becomes much slower compared to the drop in seen class accuracy ($\sim 45\%$ versus $\sim 25\%$ at around 1k iterations). This is followed by a somewhat stable plateaued trajectory for both seen and unseen class accuracies with a slight bump at around 2k iterations. After

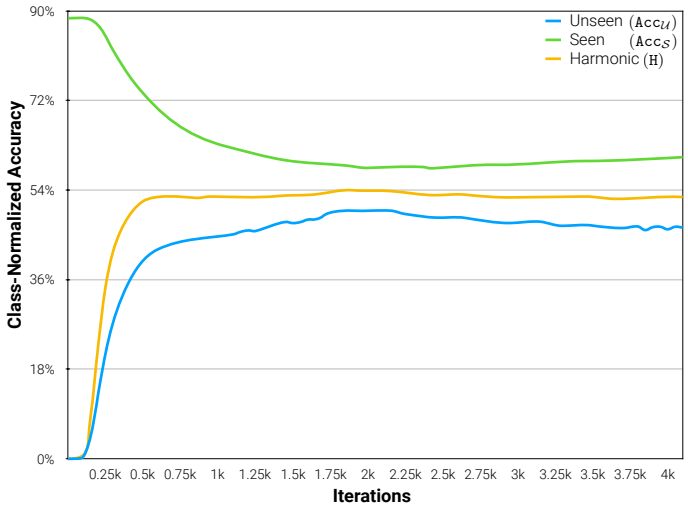


Figure 4.4: **Performance across iterations.** We study the variation in seen and unseen class normalized accuracies at different stages of the optimization process. The base architecture involved is VGG16 trained on the AWA2 dataset with the regularization coefficient set to $\lambda = 1e^{-5}$.

this, we observe significantly slower increase (decrease) in seen (unseen) class accuracies, indicating the classifier weights for the unseen classes are not perturbed significantly beyond this point to cause any drastic changes in performance. Again, as stated earlier, from the behavior of this optimization process it is clear that improvement in unseen classes (although more pronounced initially) comes at a cost of decrease in performance on the seen classes and that this cost increases very sharply within a span of $\sim$ 750 iterations. Our convergence criterion, governed by the drop in overall loss by 1% in the `next` 40 iterations stops the optimization process for this particular setting at around 450 iterations which preserves much of the seen class accuracy while gaining substantially in unseen class accuracy.

4.6 Explaining NIWT

The goal of this section is two-fold. We provide visual explanations through Grad-CAM [65] and we show how we can utilize a mapping $W_{a \to \mathcal{K}}$ from $\mathbf{a}_c$ to domain knowledge $\mathcal{K}$ to provide text-based explanations and also name semantic neurons in the network automatically.

4.6.1 Visual Explanations

Our proposed approach enables us to create an end-to-end model for novel classes, while still embedding information from different domains into the network. This enables us to directly use any of the many deep learning interpretability techniques. We use GradCAM [65] on instances of unseen classes to visualize the support for decisions made by the network with NIWT learnt classification weights. Figure 4.5 includes some sample GradCAM outputs.

Evaluating Visual Explanations:

We evaluate the generated maps for both seen and unseen classes by the mean fraction of the GradCAM activation present inside the bounding box annotation associated with the present objects. On seen-classes, we found this number to be 0.80 ± 0.008 versus 0.79 ± 0.005 for the unseen classes on CUB – indicating that the learned unseen classifier is indeed capable of focusing on relevant regions in the input.

4.6.2 Textual Explanations.

In Section 4.3.3 we computed a matrix $W_{\mathcal{K}\to a}$ that helped embed the domain knowledge in the network's last convolutional layer based on neuron importance. Similarly an inverse mapping from neuron importance to domain knowledge ($W_{a\to\mathcal{K}}$) can be learned in the context of binary attributes through a multi-label classification task. We utilize this inverse mapping to obtain scores in the attribute space and retrieve the top-k attributes as explanations. A high scoring $\mathbf{k}_c$ retrieved via $W_{a\to\mathcal{K}}$ from a certain $\mathbf{a}_c$ emphasizes the relevance of that attribute for the corresponding class c. This helps us ground the class-score decisions made by the learnt unseen classifier head in the attribute space, thus, explaining the same in the process.

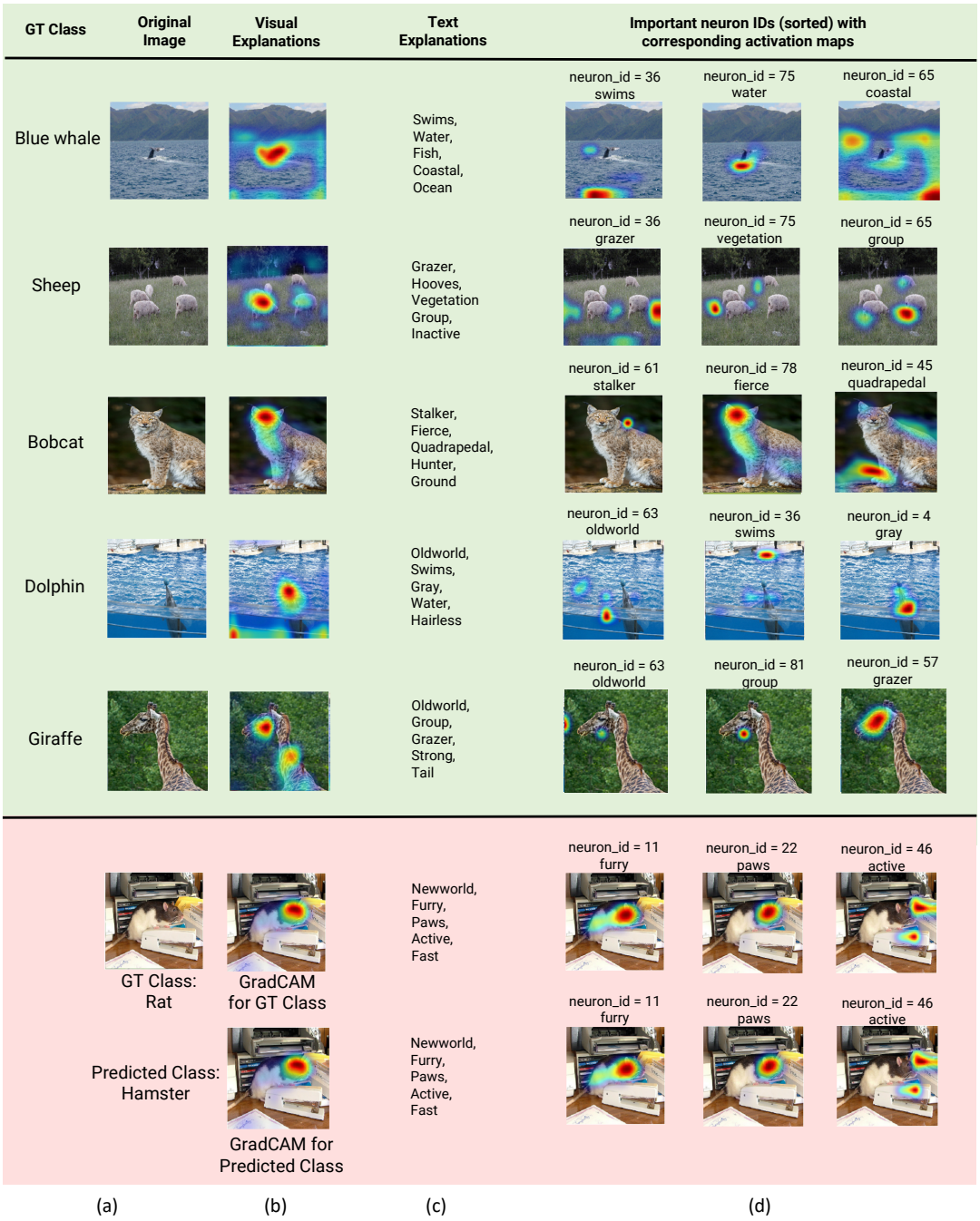


Figure 4.5: Success and failure cases for unseen classes using explanations for NIWT: Success cases: (a) the ground truth class and image, (b) Grad-CAM visual explanations for the GT category, (c) textual explanations obtained using the inverse mapping from $\mathbf{a}_c$ to domain knowledge. (d) most important neurons for this decision and neuron names, including the activation map corresponding to the neuron. The last 2 rows show negative examples, where the model predicted a wrong category. We show Grad-CAM maps and textual explanations for both the ground truth and predicted category. By looking at the explanations for the failure cases we can see that the model's mistakes are not completely unreasonable.

Evaluating Textual Explanations

We can evaluate the fidelity of such generated textual explanations by the percentage of associated ground truth attributes captured in the top-k generated explanations on a per instance level. We observe this number to be 83.9% on CUB using a VGG-16 network. Qualitative results in Fig. 4.5 shows both visual and textual explanation which show that we can get the most discriminative attribute for any given target category.

4.6.3 Neuron Names and Focus

As previously discussed, as the depth of a CNN increases, higher-level semantics are captured [59, 13, 65]. Neuron-names are referable groundings of such concepts captured by the CNN at different layers. We obtain neuron names in a cheap fashion by feeding a one-hot encoded vector corresponding to a neuron *position* to $W_{a \to \mathcal{K}}$ and perform a similar process of top-1 retrieval to obtain the corresponding 'neuron-name'. We also observe the activation map corresponding to that neuron and qualitatively evaluate whether the neurons 'focus' at the named attributes in the image.

In Fig 4.5, we provide qualitative examples for the above. The green blocks correspond to the instances where the unseen class images were correctly classified by the $\texttt{NET}_{S \cup U}$. Similarly, the red blocks correspond to the case where the image was incorrectly classified by the same. The columns correspond to the class-labels, images, Grad-CAM visualizations for the class, textual explanations in the attribute space and top-3 neuron names responsible for the target class and their corresponding activation maps. For instance, notice that in the second row, for the image - correctly classified as a yellow-headed blackbird - the visualization maps for the class look specifically at the union of attributes that comprise this class of birds. In addition, the textual explanations also filter out these attributes based on the neuron-importance scores - *has throat color yellow*, *has wing color black*, etc. In addition, when we focus on the individual neurons with relatively higher importance we see that individual neurons focus on the visual regions characterized by their 'names'.

This shows that our neuron names are indeed well grounded in the image. The fourth row, containing an instance of the groove-billed ani class is another example this occurs.

Consider the case corresponding to the misclassified example (row 7 and 8). If we look at the intersection of attribute values in the textual explanations corresponding to the ground truth and the predicted class along with the image, qualitatively we can understand why the network might be confusing the two classes as these textual explanations are grounded in the image. Similarly, the neuron names and corresponding activations have a mismatch with the predicted class with the activation maps focusing on a 'yellowish' area rather than a visual region corresponding to a fine-grained attribute.

4.7 Explanations on AWA2

4.7.1 Explanations for NIWT trained on AWA2 dataset

In this subsectionsection, we discuss explanations for the unseen classes of AWA2 [86] under the proposed split. In total, there are 10 unseen classes. Similar to Fig. 4 in the main paper, Fig. 4.5 shows similar examples on the unseen classes of AWA2. Note that the attributes in the AWA2 dataset are much less fine-grained and visually grounded compared to CUB [100] and hence, while the retrieved neuron-names in this case are feasible attributes associated with the class concerned, neuron focus is often harder to interpret (for instance, the activation map associated with the attribute *old world* is arbitrary).

We observe that for the success cases (in green), Grad-CAM [65] visualizations corresponding to the concerned class are focused on the object of interest in the image irrespective of the amount of saliency of the class present. However, the neuron focus is heavily dependent on the size of the class present in the image and how visually grounded the associated attributes for that class are. For instance, in the 3rd row, the focus associated with he neurons 78 (fierce) and 45 (quadrapedal) is interpretable and visually grounded in the *bobcat* present. However, in the 5th row, we notice that although the retrieved neuron names are associated with the class *giraffe*, the activation maps associated with each of

these neurons (63, 81 and 57) are not entirely feasible. Interestingly, in the misclassified example, where a *rat* is mistaken for a *hamster*, we can attribute the misclassification to the inability of $\texttt{NET}_{\mathcal{S}+\mathcal{U}}$ in this case, to focus on a discriminative attribute associated with a *rat* and a *hamster*. This in turn also motivates the idea to ground the neuron-importances in some domain representation that are not only definitive of a class but are discriminative relative to other classes as well.

4.8 Conclusion

To summarize, in this chapter we proposed an approach we refer to as Neuron Importance-aware Weight Transfer (NIWT), that learns to map domain knowledge about novel classes directly to classifier weights by grounding it into the importance of network neurons. Our weight optimization approach on this grounding results in classifiers for unseen classes which outperform existing approaches at a popular generalized zero-shot learning benchmark. We further demonstrate that this grounding between language and neurons can also be learned in reverse, linking neurons to human interpretable semantic concepts.

CHAPTER 5

TAKING A HINT: LEVERAGING EXPLANATIONS TO MAKE VISION AND LANGUAGE MODELS MORE GROUNDED

5.1 Introduction

Many popular and well-performing models for multi-modal, vision-and-language tasks exhibit poor visual grounding – failing to appropriately associate words or phrases with the image regions they denote and relying instead on superficial linguistic correlations [7, 2, 32, 34, 103]. For example, answering the question *'What color are the bananas?'* with yellow regardless of their ripeness evident in the image. When challenged with datasets that penalize reliance on these sort of biases [7, 34], state-of-the-art models demonstrate significant drops in performance despite there being no change to the set of visual and linguistic concepts about which models must reason.

In addition to these diagnostic datasets, another powerful class of tools for observing this shortcoming has been gradient-based explanation techniques [104, 63, 105, 106] which allow researchers to examine which portions of the input models rely on when making decisions. Application of these techniques has shown that vision-and-language models often focus on seemingly irrelevant image regions that differ significantly from where human subjects fixate when asked to perform the same tasks [35, 65] – *e.g.* focusing on a produce stand rather than the bananas in our example.

While somewhat dissatisfying, these findings are not entirely surprising – after all, standard training protocols do not provide any guidance for visual grounding. Instead, models are trained on input-output pairs and must resolve grounding from co-occurrences – a challenging task, especially in the presence of more direct and easier to learn correlations in language. Consider our previous example question, the words 'color', 'banana', and 'yel-

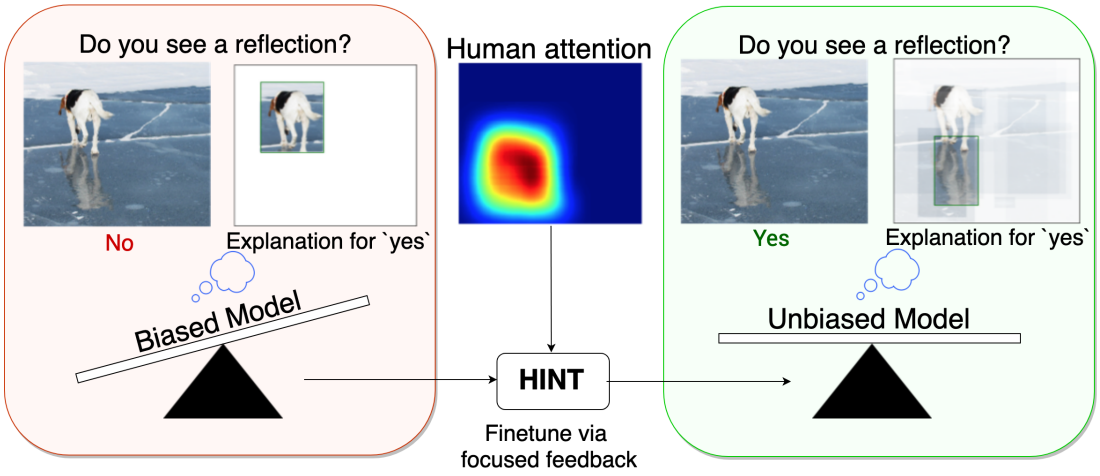


Figure 5.1: Our approach, HINT, aligns visual explanations for output decisions of a pretrained model with spatial input regions deemed important by human annotators – forcing models to base their decisions on these same region and reducing model bias.

low' are given as discrete tokens that will trivially match in every occurrence when these underlying concepts are referenced. In contrast, actually grounding this question requires dealing with all visual variations of bananas and learning the common feature of things described as 'yellow'. To address this, we explore if giving a *small hint* in the form of human attention demonstrations can help improve grounding and reliability.

For the dominant paradigm of vision-and-language models that compute an explicit question-guided attention over image regions [25, 26, 27, 28, 29, 30], a seemingly straightforward solution is to provide explicit grounding supervision – training models to attend to the appropriate image regions. While prior work [38, 107] has shown this approach results in more human-like attention maps, our experiments show it to be ineffective at reducing language bias. Crucially, attention mechanisms are bottom-up processes that feed final classification models such that *even when attending to appropriate regions, models can ignore visual content in favor of language bias.* In response, we introduce a generic, second-order approach that instead aligns gradient-based explanations with human attention.

Our approach, which we call Human Importance-aware Network Tuning (HINT), enforces a ranking loss between human annotations of input importance and gradient-based explanations produced by a deep network – updating model parameters via a gradient-of-

gradient step. Importantly, this constrains models to not only look at the correct regions but to also be sensitive to the content present there when making predictions. While we experiment with HINT in the context of vision-and-language problems, the approach itself is general and can be applied to focus model decisions on specific inputs in any context.

We apply HINT to two tasks – Visual Question Answering (VQA) [1] and image captioning [108] – and find our approach significantly improves visual grounding. With human importance supervision for only 6% of the training set, our HINT'ed model improves the state-of-the-art by 8 percentage points on the challenging dataset VQA Under Changing Priors (VQA-CP) [7], which is designed to test visual grounding. In both VQA and Image Captioning, we see significantly improved correlations between human attention and visual explanations for HINT trained models, showing that models learn to make decisions using similar evidence as humans (even on new images). We perform human studies which show that humans perceive models trained using HINT to be more reasonable and trustworthy.

Contributions. To summarize our contributions, we

- introduce Human Importance-aware Network Tuning (HINT), a general approach for constraining the sensitivity of deep networks to specific input regions and demonstrate it results in significantly improved visual grounding for two vision and language tasks,
- set a new state-of-the-art on the bias-sensitive VQA Under Changing Priors (VQA-CP) dataset [7], and
- conduct studies showing that humans find HINTed models more trustworthy than standard models.

5.2 Related Work

Model Interpretability. There has been significant recent interest in building machine learning models that are transparent and interpretable in their decision making process. For deep networks, several works propose explanations based on internal states of the network [10, 109, 69, 106]. Most related to our work is the approach of Selvaraju *et al*. [106] which

computes neuron importance as part of a visual explanation. In this work, we enforce that these importance scores align with importances provided by domain experts.

Vision and Language Tasks. Image Captioning [24] and Visual Question Answering (VQA) [1] have emerged as two of the most widely studied vision-and-language problems. The image captioning task requires generating natural language descriptions of image contents and the VQA task requires answering free-from questions about images. In both, models must learn to associate image content with natural free-form text. Consequentially, attention based models that explicitly reason about image-text correspondences have become the dominant paradigm [25, 26, 27, 28, 29, 30]; however, there is growing evidence that even these attentional models still latch onto language biases [7, 32, 33].

Recently, Agrawal *et al*. [7] introduced a novel, bias-sensitive dataset split for the VQA task. This split, called VQA Under Changing Priors (VQA-CP), is constructed such that the answer distributions differ significantly between training and test. As such, models that memorize language associations in training instead of actually grounding their answers in image content will perform poorly on the test set. Likewise Lu *et al*. [29] introduce a robust captioning split of the COCO captioning dataset [24] in which the distribution of co-occurring objects differs significantly between training and test. We use these dataset splits to evaluate the impact of our method on visual grounding.

Debiasing Vision and Language Models. A number of recent works have aimed to reduce the effect of language bias in vision and language models.

Hendricks *et al*. [33] study the generation of gender-specific words in image captioning – showing that models nearly always associated male gendered words to people performing extreme sports like snowboarding regardless of the image content. Their presented Equalizer approach encourages models to adjust their confidence depending on the evidence present – confident when gender evidence is visible and unsure when it is occluded by ground-truth segmentation masks. Experiments on a set of captions containing people show this approach reduces gender bias.

For VQA, Agrawal *et al*. [7] developed a Grounded VQA model (GVQA) that disentangles the vision and language components – consisting of separate visual concept and answer cluster classifiers. This approach uses a question's type (*e.g*. "What color ...") to determine the space of possible answers and the question target (*e.g*. "banana") to detect visual attributes in the scene that are then filtered by the possible answer set. While effective, this requires multi-stage training and is difficult to extend to new models. Ramakrishnan *et al*. [110] introduce an adversarial model agnostic regularization technique to reduce bias in VQA models – pitting the model against a question-only adversary.

Human Attention for VQA. Das *et al*. [35] collected human attention maps for a subset of the VQA dataset [1]. Given a question and a blurry image, humans were asked to interactively deblur regions in the image until they could confidently answer. In this work, we utilize these maps, enforcing the gradient-based visual explanations of model decisions to closely match the human attention.

Supervising model attention. Liu *et al*. [107] and Qiao *et al*. [38] apply human attention supervision to attention maps produced by the model for image captioning and VQA, respectively. We experiment with a similar approach but find that the improved attention correlation does not translate to reduced reliance on language bias – even with appropriate model attention, the remaining network layers can still disregard the visual signal in the presence of strong biases. We also show how gradient explanations are more faithful to model decisions by directly linking model decisions input regions, so that aligning these importances ensures the model is basing its decision on human-attended regions.

Aligning gradient-based importances. Selvaraju *et al*. [111] proposed an approach to learn a mapping between gradient-based importances of individual neurons within a deep network (from [106]) and class-specific domain knowledge from humans in order to learn classifiers for novel classes. In contrast, we align gradient-based importances to human attention maps to improve network grounding.

5.3 Preliminaries

While our approach is general-purpose and model agnostic, in this work we take the recent Bottom-up Top-down architecture [30] as our base model. A number of works [112, 45, 113, 114, 115, 28, 73] use Top-down attention mechanisms to help fine-grained and multi-stage reasoning, which is shown to be very important for vision and language tasks. Anderson *et al*. [30] propose a variant of the traditional attention mechanism, where instead of attending over convolutional features they show that attending over objects and other salient image regions gives significant improvements in VQA and captioning performance. We briefly describe this architecture below, see [30] for full details.

Bottom-Up Top-Down Attention for VQA. As shown in left half of Fig. 5.2, given an image, the Bottom-up Top-down (UpDown) attention model takes as input up to k image features, each encoding a salient image region. These regions and their features are proposals extracted from Faster-RCNN [116]. The question is encoded using a GRU [117] and a soft-attention over each of the k proposal features is computed using the question embedding. The final pooled attention feature is combined with the question feature using a few fully-connected layers which predict the answer.

Bottom-Up Top-Down Attention for Image Captioning. The image captioning model consists of two Long Short-Term Memory (LSTM) networks – an attention LSTM and a language LSTM. The first LSTM layer is a top-down visual attention model whose input at each time step consists of the previous hidden state of the language LSTM, concatenated with the mean-pooled bottom-up proposal features (similar to above) and an encoding of the previously generated word. The output of the attention LSTM does a soft attention over the proposal features. The second LSTM is a language generation LSTM that takes as input the attended features concatenated with the output of the attention LSTM. The language LSTM provides a distribution over the vocabulary of words for the next time step.

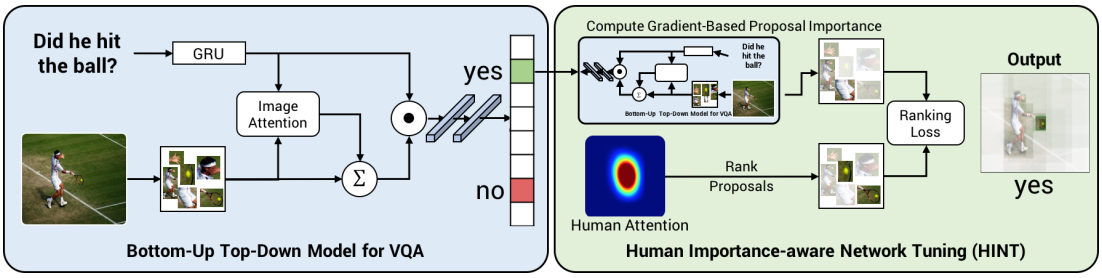


Figure 5.2: Our Human Importance-aware Network Tuning (HINT) approach: Given an image and a question like "Did he hit the ball?", we pass them through the Bottom-up Top-down architecture shown in the left. For the example shown, the model incorrectly answers 'no'. We determine the proposals important for the ground-truth answer 'yes' through a gradient-based importance measure. We rank the proposals through human attention and provide a ranking loss in order to align the network's importance with human importance. Tuning the model through HINT makes the model not only answer correctly, but also look at the right regions, as shown in the right.

5.4 Human Importance-aware Network Tuning

In this section, we describe our approach for training deep networks to rely on the same regions as humans which we call Human Importance-aware Network Tuning (HINT). In summary, HINT estimates the importance of input regions through gradient-based explanations and tunes the network parameters so as to align this with the regions deemed important by humans. We use the generic term 'prediction' to refer to both answers in the case of VQA and the words generated at each time step in image captioning.

5.4.1 Human Importance

In this step, we align the expert knowledge obtained from humans attention maps into a form corresponding to the network inputs. The Bottom-up Top-down model [30] takes in as input region proposals. For a given instance, we compute an importance score for each of the proposals based on normalized human attention map energy inside the proposal box relative to the normalized energy outside the box.

More concretely, consider a human importance map $A^d \in \mathbb{R}^{h \times w}$ that indicates the spatial regions of support for an output d[1] – a high value $A^d[i, j]$ indicates high support for d at location (i,j). Given a proposal region r with area a_r, we can write the normalized

[1]For VQA, these maps will vary across questions for a given image.

importance inside and outside r for decision d as

$$E_i^d(r)=\frac{1}{a_r}\sum_{(i,j)\in r} A_{ij}^d \text{ and } E_o^d(r)=\frac{1}{h.w-a_r}\sum_{(i,j)\notin r} A_{ij}^d$$

respectively. We compute the overall importance score for proposal k for decision d as:

$$s_k^d = \frac{E_i^d(k)}{E_i^d(k)+E_o^d(k)} \tag{5.1}$$

Human attention for VQA and captioning. For VQA, we use the human attention maps collected by Das *et al*. [118] for a subset of the VQA [1] dataset. HAT maps are available for a total of 40554 image-question pairs – *or approximately only ∼6% of the VQA dataset.* While human attention maps do not exist for image captioning, COCO dataset [108] has segmentation annotations for 80 everyday occurring categories. We use a word-to-object mapping that links fine-grained labels like ["child", "man", "woman", ...] to object categories like <person> similar to [29]. We map a total of 830 visual words existing in COCO captions to 80 COCO categories. We then use the segmentation annotations for the 80 categories as human attention for this subset of matching words. To be consistent with the VQA setup, we only use 6% of the segmentation annotations.

5.4.2 Network Importance

We define Network Importance as the importance that the given trained network places on spatial regions of the input when making a particular prediction. Selvaraju *et al*. [106] proposed an approach to compute the importance of last convolutional layer's neurons. In their work, they focus on the last convolutional layer neurons as they serve as the best compromise between high level semantics and detailed spatial information. Since proposals usually look at objects and salient/semantic regions of interest while providing a good spatial resolution, we extend [65] to compute importance over proposals. In order to obtain the importance of a proposal r for ground-truth decision, α_{gt}^r, we one-hot encode the score

for the ground-truth output (answer in VQA and the visual word in case of captioning) o_{gt} and compute its gradients w.r.t. proposal features as,

$$\alpha_{gt}^{r} = \overbrace{\sum_{i=1}^{|P|}}^{\text{global pooling}} \underbrace{\frac{\partial o_{gt}}{\partial P_i^r}}_{\text{gradients via backprop}} \tag{5.2}$$

Note that we compute the importance for the ground-truth decision, and not predicted. Human attention for incorrect decisions are not available and are conceptually ill-posed because it is difficult to define what correct 'evidence' for an incorrect prediction would be.

5.4.3 Human-Network Importance Alignment

At this stage, we now have two sets of importance scores – one computed from the human attention and another from network importance – that we would like to align. Each set of scores is calibrated within itself; however, absolute values are not comparable between the two as human importance lies in $[0, 1]$ while network importance is unbounded. Consequentially, we focus on the relative rankings of the proposals, applying a ranking loss – specifically, a variant of Weighted Approximate Rank Pairwise (WARP) loss.

Ranking loss. At a high level, our ranking loss searches all possible pairs of proposals and finds those pairs where the pair-wise ranking based on network importance disagrees with the ranking from human importance. Let $\mathcal{S}$ denote the set of all such misranked pairs. For each pair in $\mathcal{S}$, the loss is updated with the absolute difference between the network importance score for the proposals pair.

$$\mathcal{L} = \sum_{(r',r)\in\mathcal{S}} \left| \alpha_{-}^{r'} - \alpha_{+}^{r} \right| \tag{5.3}$$

where r and r' are the proposals whose order based on neuron importance does not align with human importance and $+$ indicates that proposal r is more important compared to r'

according to human importance.

Importance of task loss. In order to retain performance at the base task, it is necessary to include the original task loss λL_{Task} – cross-entropy for VQA and negative log-likelihood in case of image captioning. To trade-off between the two, we introduce a multiplier λ such that the final HINT loss becomes,

$$\mathcal{L}_{\mathcal{HINT}} = \sum_{(r',r)\in\mathcal{S}} \left|\alpha_{-}^{r'} - \alpha_{+}^{r}\right| + \lambda L_{Task} \tag{5.4}$$

The first term encourages the network to base predictions on the correct regions and the second term encourages it to actually make the right prediction.

Note that network importances α are gradients of the score with respect to proposal embeddings. Thus they are a function of all the intermediate parameters of the network ranging from the model attention layer weights to the final fully-connected layer weights. Hence an update through an optimization algorithm (gradient-descent or Adam) with the given loss in (5.4) requires computation of second-order gradients, and would affect all the network parameters. We use PyTorch [119] which has this functionality.

5.5 Experiments and Analysis

In this section we describe the experimental evaluation of our approach on VQA and Image Captioning.

VQA. For VQA, we evaluate on the standard VQA split and the VQA-CP [7] split. Recall from Section 5.2 that VQA-CP is a restructuring of VQAv2 [34] that is designed such that the answer distribution in the training set differs significantly from that of the test set. For example, while the most popular answer in train for "What sport ..." questions might be "tennis", in test it might be "volleyball". Without proper visual grounding, models trained on this dataset will generalize poorly to the test distribution. In fact, [7] and [110] report significant performance drops for state-of-the-art VQA models on this challenging, language-bias sensitive split. For our experiments, we pretrain our Bottom-Up Top-Down

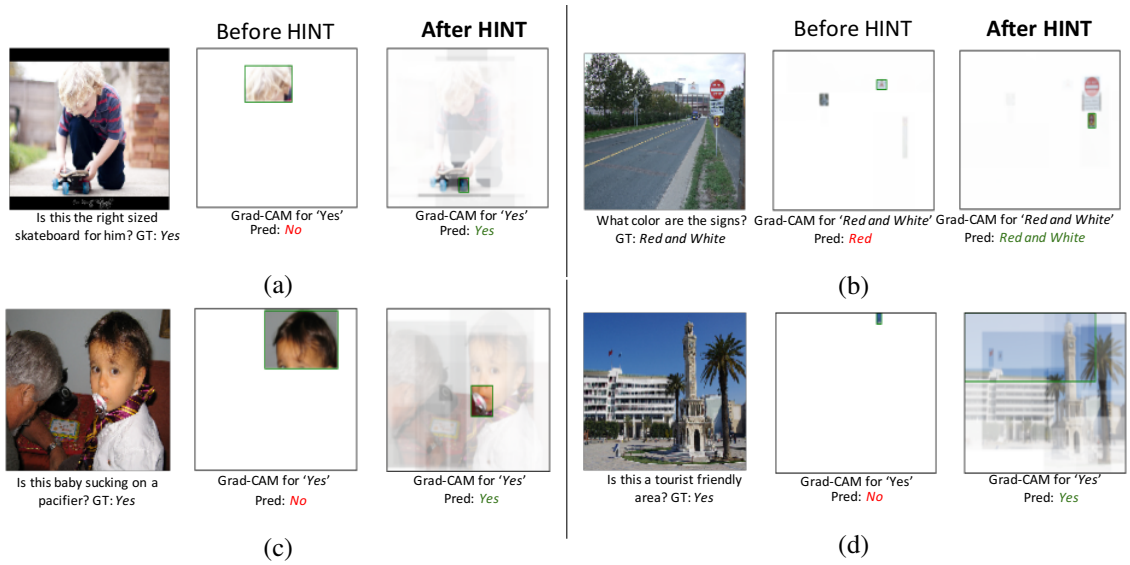


Figure 5.3: Qualitative comparison of models on validation set before and after applying HINT. For each example, the left column shows the input image along with the question and the ground-truth (GT) answer from the VQA-CP val split. In the middle column, for the base model we show the explanation visualization for the GT answer along with the model's answer. Similarly we show the explanations and predicted answer for the HINTed models in the third column. We see that the HINTed model looks at more appropriate regions and answers more accurately. For example, for the example in (a), the base model only looks at the boy, and after we apply HINT, it looks at both the boy and the skateboard in order to answer 'Yes'. After applying HINT, the model also changes its answer from 'No' to 'Yes'. More qualitative examples can be found in the supplementary material.

model on respective training splits before fine-tuning with the HINT loss. Recall that our approach includes the task loss; We use $\lambda_{vqa} = 10$ for our experiments.

We compare our approach against strong baselines and existing approaches, specifically:

- **Base Model (UpDn)** We compare to the base Bottom-up Top-down model without our HINT loss.

- **Attention Alignment (Attn. Align.)** We replace gradient supervision with attention supervision keeping everything else the same. The Bottom-up Top-down model uses soft attention over object proposals – essentially predicting a set of attention scores for object proposals based on their relevancy to the question. These attention scores are much like the network importances we compute in HINT; however, they are functions only of the network prior to attention prediction. We apply the HINT

Table 5.1: Results on compositional (VQA-CP) and standard split (VQAv2). We see that our approach (HINT) gets a significant boost of over 7% from the base UpDn model on VQA-CP and minor gains on VQAv2. The Attn. Align baseline sees similar gains on VQAv2, but fails to improve grounding on VQA-CP. Note that for VQAv2, during HINT finetuning we apply the VQA cross entropy loss even for the samples without human attention annotation. † results taken from corresponding papers.

Model	VQA-CP `test`				VQAv2 `val`			
	Overall	Yes/No	Number	Other	Overall	Yes/No	Number	Other
SAN [28]	24.96	38.35	11.14	21.74	52.41	70.06	39.28	47.84
UpDn [30]	39.49	45.21	11.:96	42.98	62.85	80.89	42.78	54.44
GVQA [7]†	31.30	57.99	13.68	22.14	48.24	72.03	31.17	34.65
UpDn + Attn. Align	39.37	43.02	11.89	45.00	63.24	80.99	42.55	55.22
UpDn + AdvReg [110]†	41.17	65.49	**15.48**	35.48	62.75	79.84	42.35	55.16
UpDn + HINT (ours)	**46.73**	**67.27**	10.61	**45.88**	**63.38**	**81.18**	**42.99**	**55.56**

ranking loss between these attention weights and human importances as computed in Equation (5.1).

- **Grounded VQA (GVQA).** As discussed in Section 5.2, [7] introduced a grounded VQA model that explicitly disentangles vision and language components and was developed alongside the VQA-CP dataset.

- **Adversarial Regularization (AdvReg).** [110] introduced an adversarial regularizer to reduce the effect of language-bias in VQA by explicitly modifying question representations to fool a question-only adversary model.

Image Captioning. For captioning, we evaluate on the standard 'Karpathy' split and the robust captioning split introduced by Lu *et al*. in [29]. The robust split has varying distribution of co-occurring objects between train and test. We pretrain our Bottom-up Top-down captioning model on the respective training splits and apply our approach, HINT. Note that the HINT loss is applied only for the time steps corresponding to the 830 visual words in the caption that we obtain in Section 5.4.1.

5.5.1 HINT for Visual Question Answering

Table 5.1 shows results for our models and prior work on VQA-CP test and VQAv2 val. We summarize key results:

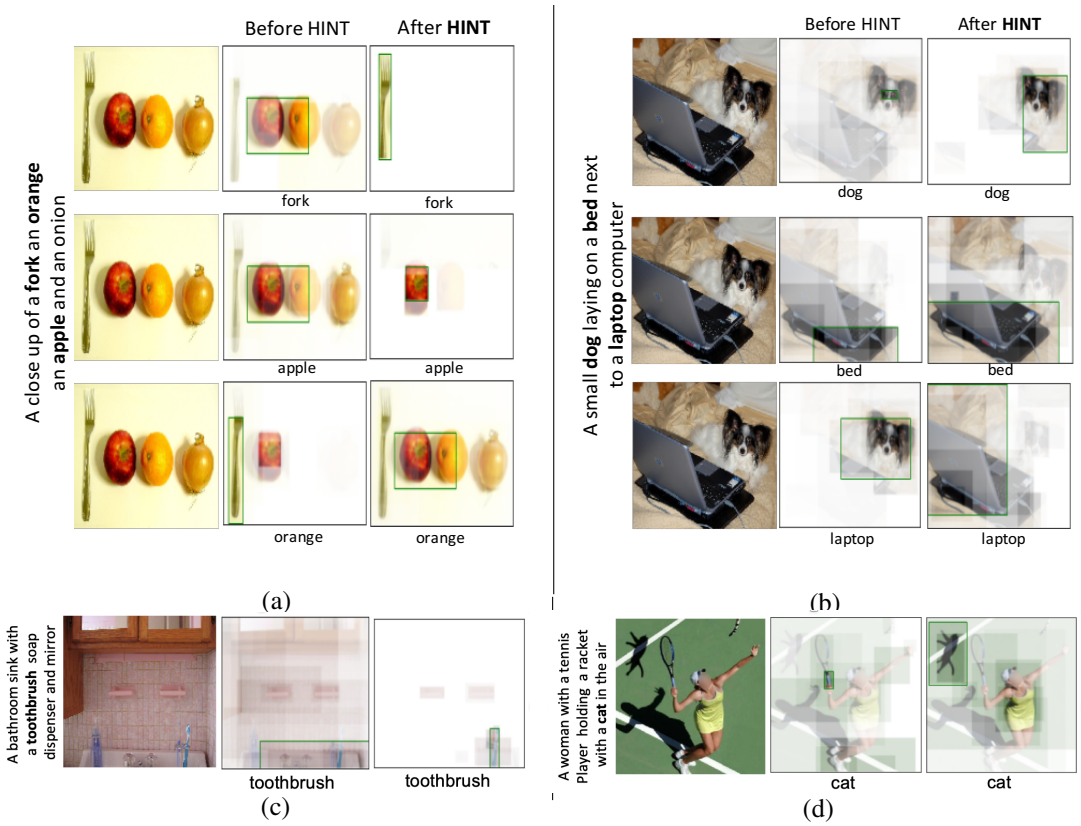


Figure 5.4: Qualitative comparison of captioning models on validation set before and after applying HINT. For each example, the left column shows the input image along with the ground-truth caption from the COCO robust split. In the middle column, for the base model we show the explanation visualization for the visual word mentioned below. Similarly we show the explanations for the HINTed models in the third column. We see that the HINTed model looks at more appropriate regions. For example in (a) note how the HINTed model correctly localizes the fork, apple and the orange when generating the corresponding visual words, but the base model fails to do so. Interestingly the model is able to ground even the shadow of a cat in (f)! More qualitative examples can be found in the supplementary material.

HINT reduces language-bias. For VQA-CP, our HINTed UpDown model significantly improves over its base architecture alone by 7 percentage point gain in overall accuracy. Further, it outperforms existing approaches based on the same UpDn architecture (41.17 for AdvReg vs 46.73 for HINT), setting a new state-of-the-art for this problem. We do note that our approach uses additional supervision in the form of human attention maps for 6% of training images.

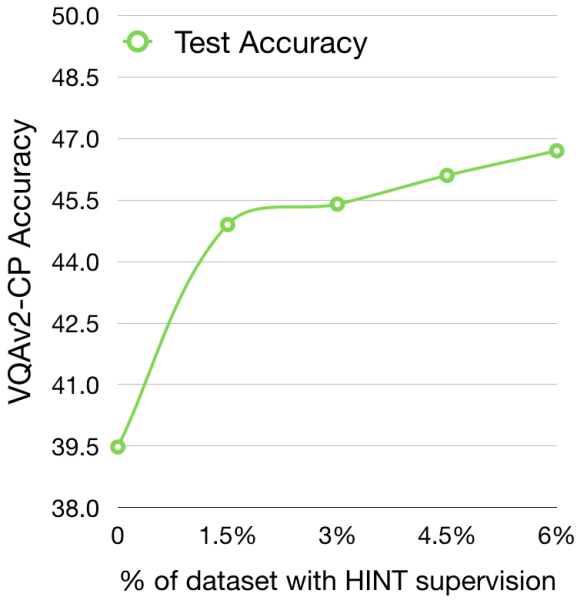
HINT improves grounding without reducing standard VQA performance. Unlike previous approaches for language-bias reduction which cite trade-offs in performance between

the VQA and VQA-CP splits [110, 7], we find our HINTed UpDn model actually improves on standard VQA – making HINT the first ever approach to show simultaneous improvement on both the standard and compositional splits.

Note: Given that human attention is only available for 6% of the dataset, it is possible that aligning network attention with human attention on this limited data only provides regularization benefits. One way to confirm this hypothesis could be by aligning network attention with random human attention targets. If improvements are observed even in that scenario, this would indicate that HINT mostly provides optimization regularization related benefits.

Attn. Align is ineffective compared to HINT. A surprising (to us at least) finding and motivating observation of this work is that directly supervising model attention (as in Attn. Align) is ineffective at reducing language-bias and improving visual grounding as measured by VQA-CP, begging the question – why does our gradient supervision succeed where attention supervision fails?

We argue this results from gradient-based explanations being 1) a function of all network parameters unlike attention alignment and 2) more faithful to model decisions than model attention. As we've discussed previously, attention is a bottom-up computation and supervising it cannot directly affect later network layers, whereas our HINT approach does. To assess faithfulness, we run occlusion studies similar to those in [106, 10]. We measure the difference in model scores for the predicted answer when different proposal features for the image are masked and forward propagated, taking this delta as an importance score for each proposal. We find that rank correlation between model attention and occlusion-based importance is only 0.10, compared to 0.48 for gradient-based importance – demonstrating our claim that *model attention only loosely relates to how the model actually arrives at its decision.* As such, attention alignment simply requires the model to predict human-like attention, not necessarily to care about them when making decisions. On the other hand, HINT aligns gradient-based importance with respect to model decisions, ensuring that hu-



man specified regions are actually used by the network – resulting in a model that is *right for the right reasons*.

Varying the amount of human attention supervision. The plot below shows performance for different amounts of Human Attention maps for VQA-CP. Note that the x-axis goes from using no HINT supervision to using all the Human attention maps during training, which amounts to 6% of the VQAv2 data. Note that with human attention supervision for just 1.5% of the VQA dataset, our approach achieves a 5 % improvement in performance.

Qualitative examples. Fig. 5.3 shows qualitative examples showing the effect of applying HINT to the Bottom-up Top-down VQA model. Fig. 5.3 (b) shows an image and a question, *'What color are the signs?'*, the base model answers "Red" which is partially correct, but it fails to ground the answer correctly. The HINTed model not only answers "Red and White" correctly but also looks at the red stop sign and the white street sign.

5.5.2 HINT for Image Captioning

Our implementation of the Bottom-up Top-down captioning model in Pytorch [119] achieves a CIDEr [120] score of 1.06 on the standard split and 0.90 on the robust split. Upon applying HINT to the base model trained on the robust split, we obtain a CIDEr score of 0.92, an improvement of 0.02 over the base model. For the model trained on the standard split, performance drops by 0.02 in CIDEr score (1.04 compared to 1.06). As we show in the following sections, the lack of improvement in score does not imply a lack of change – we find the model shows significant improvements at grounding, which we evaluate in Section 5.6. Note that our setup for captioning does not require *task-specific human attention*, and instead allows us to directly leverage existing annotations which were collected for a different task (image segmentation).

Qualitative examples. Fig. 5.4 shows qualitative examples that indicate significant improvements in grounding performance of HINTed models. For example Fig. 5.4 (a) shows how a model trained with HINT is able to simultaneously improve grounding for the 3 visual words present in the ground-truth caption. We see that HINT also helps with making models focus on individual object occurrences rather than using context, as shown in Fig. 5.4 (c, d, e, f).

5.6 Evaluating Grounding

In Sections 5.5.1 and 5.5.2 we evaluated the effect of HINT on the task performance, with generalization to robust dataset splits serving as an indirect evaluation of grounding. In this section we directly evaluate the grounding ability of models tuned with HINT.

5.6.1 Correlation with Human Attention

In order to evaluate the grounding ability of models before and after applying HINT, we compare the network importances for the ground-truth decision (as in Equation (5.2)) with

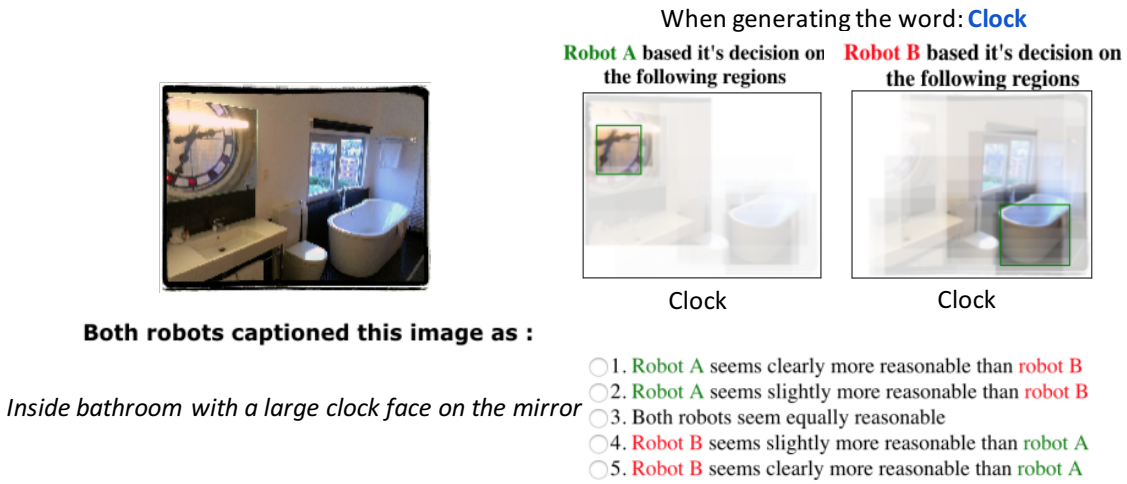


Figure 5.5: AMT interface for evaluating the baseline captioning model and our HINTed model. HINTed model outperforms baseline model in terms of human trust.

the human attention as computed in Equation (5.1) for both the base model and the model fine-tuned with HINT. We then compute the rank correlation between the network importance scores and human importance scores for images from the VQA-CP and COCO robust test splits. We report Spearman's rank correlation between explanations from the base model and the HINTed model.

VQA. For the model trained on VQA-v2, we find that the Grad-CAM based attention for base model obtains a Spearman's rank correlation of -0.09 with human attention maps [118]. Note that the range of rank-correlation is -1 to 1, so near 0 indicates no correlation. We find that the HINTed model obtains a correlation of 0.18.

Image Captioning. For the model trained on the COCO robust split, the Grad-CAM based attention for base model achieves a rank correlation of 0.008 with COCO segmentation maps for the visual words, and the model after HINTing achieves a correlation of 0.17.

This rank correlation measure matches the intent of the rank-based HINT loss, but this result shows that the visual grounding learned during training generalizes to new images and language contexts better than the baseline model.

5.7 Evaluating Trust

In the previous section we evaluate if HINTed models attend to the same regions as humans when forced into making predictions. Having established that, we turn to understanding whether this improved grounding translates to increased human trust in HINTed models. We focus this study on our image captioning models.

We conduct human studies to evaluate if based on individual prediction explanations from two models – the base model and one with improved grounding through HINT – humans find either of the models more trustworthy. In order to tease apart the effect of grounding from the accuracy of the models being visualized, we only visualize predictions corresponding to the ground-truth caption for both models.

For a given ground truth caption, we show study participants the network importance explanation for a ground truth visual word as well as the whole caption. Workers were then asked to rate the reasonableness of the models relative to each other on a 5-point Likert scale of clearly more/less reasonable (+/-2), slightly more/less reasonable (+/-1), and equally reasonable (0). This interface is shown in Fig. 5.5. In order to eliminate any biases, the base and HINTed models were assigned to be 'model1' with equal probability.

In total, 42 Amazon Mechanical Turk (AMT) workers participated in the study, producing 1000 responses (5 annotations corresponding to 200 image pairs). In 49.9 % of instances, participants preferred HINT compared to only 33.1 % for the base model. These results indicate that HINT helps models look at appropriate regions, and that this in turn makes the model more trustworthy.

5.8 Does HINT also improve model attention?

While HINT operates on answer gradient maps, we find it also improves feed-forward model attention. For VQA, we compute IoU of the top scoring proposal box with the human attention maps from Park *et al*. 2018. UpDn trained on VQA-CP obtained an IoU

of 0.57 whereas after applying HINT we achieve an IoU of 0.63.

We conduct human studies (similar to Section 5.7) to evaluate trust based on model attention. We collected 10 responses each for 100 randomly sampled image-question pairs. 31% of respondents found HINTed VQA-CP model to be more trustworthy compared to 16.5% for the base model. This was not the primary objective of our approach but is a promising outcome for feed-forward attention!

5.9 Conclusion

We presented Human Importance-aware Network Tuning (HINT), a general framework for aligning network sensitivity to spatial input regions that humans deemed as being relevant to a task. We demonstrated this method's effectiveness at improving visual grounding in vision and language tasks such as VQA and Image Captioning. We also show that better grounding not only improves the generalization capability of models to changing test distributions, but also improves the trust-worthiness of model.

Taking a broader view, the idea of regularizing network gradients to achieve desired computational properties (grounding in our case) may prove to be more widely applicable to problems outside of vision and language – enabling users to provide focused feedback to networks.

CHAPTER 6
SQUINTING AT VQA MODELS: INTROSPECTING VQA MODELS WITH SUB-QUESTIONS

6.1 Introduction

Human cognition is thought to be compositional in nature: the visual system recognizes multiple aspects of a scene which are combined into shapes [121] and understandings. Likewise, complex linguistic expressions are built from simpler ones [122]. Similarly, tasks like Visual Question Answering (VQA) require models to perform inference at multiple levels of abstraction. For example, to answer the question, "Is the banana ripe enough to eat?" (Figure 6.1), a VQA model has to be able to detect the bananas and extract associated properties such as size and color (perception), understand what the question is asking, and reason about how these properties relate to known properties of edible bananas (ripeness) and how they manifest (yellow versus green in color). While "abstraction" is complex and spans distinctions at multiple levels of detail, we focus on separating questions into Perception and Reasoning questions. Perception questions only require visual perception to recognize existence, physical properties or spatial relationships among entities, such as "What color is the banana?" or "What is to the left of the man?", while Reasoning questions require the composition of multiple perceptual tasks and knowledge that harnesses logic and prior knowledge about the world, such as "Is the banana ripe enough to eat?".

Current VQA datasets [1, 34, 123] contain a mixture of Perception and Reasoning questions, which are considered equivalent for the purposes of evaluation and learning. Categorizing questions into Perception and Reasoning promises to promote a better assessment of visual perception and higher-level reasoning capabilities of models, rather than conflating these capabilities. Furthermore, we believe it is useful to identify the Percep-

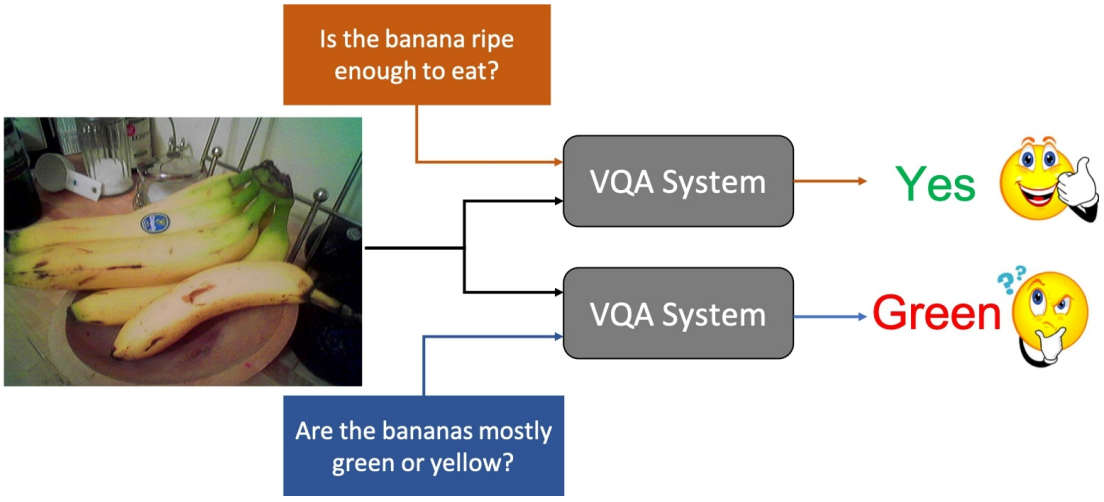


Figure 6.1: A potential reasoning failure: Current models answer the Reasoning question "Is the banana ripe enough to eat?" correctly with the answer "Yes". We might assume that doing so stems from perceiving relevant concepts correctly – perceiving yellow bananas in this example. But when asked "Are the bananas mostly green or yellow?", the model answers the question incorrectly with "Green" – indicating that the model possibly answered the original Reasoning question for the wrong reasons even if the answer was right. We quantify the extent to which this phenomenon occurs in VQA and introduce a new dataset aimed at stimulating research on well-grounded reasoning.

tion questions that serve as subtasks in the compositional processes required to answer the Reasoning question. By elucidating such "sub-questions," we can check whether the model is reasoning appropriately or if it is relying on spurious shortcuts and biases in datasets [7]. For example, we should be cautious about the model's inferential ability if it simultaneously answers "no" to "Are the bananas edible?" and "yellow" to "What color are the bananas?", even if the answer to the former question is correct. The inconsistency between the higher-level reasoning task and the lower-level perception task that it builds upon suggests that the system has not learned effectively how to answer the Reasoning question and will not be able to generalize to same or closely related Reasoning question with another image. The fact that these sub-questions are in the same modality (i.e. questions with associated answers) allows for the evaluation of any VQA model, rather than only models that are trained to provide justifications. It is this key observation that we use to develop an evaluation methodology for Reasoning questions.

The dominant learning paradigm for teaching models to answer VQA tasks assumes that models are given <image, question, answer> triplets, with no additional annotation

on the relationship between the question and the compositional steps required to arrive at the answer. As reasoning questions become more complex, achieving good coverage and generalization with methods used to date will likely require a prohibitive amount of data. Alternatively, we employ a hierarchical decomposition strategy, where we identify and link Reasoning questions with sets of appropriate Perception sub-questions. Such an approach promises to enable new efficiencies via compositional modeling, as well as lead to improvements in the consistency of models for answering Reasoning questions. Explicitly representing dependencies between Reasoning tasks and the corresponding Perception tasks also provides language-based grounding for reasoning questions where visual grounding [38, 31] may be insufficient, e.g., highlighting that the banana is important for the question in Figure 6.1 does not tell the model how it is important (i.e. that color is an important property rather than size or shape). Again, the fact that such grounding is in question-answer form (which models already have to deal with) is an added benefit. Such annotations allow for attempts to enforce reasoning devoid of shortcuts that do not generalize, or are not in line with human values and business rules, even if accurate (e.g. racist behavior).

We propose a new split of the VQA dataset, containing only Reasoning questions (defined previously). Furthermore, for questions in the split, we introduce VQA-Introspect, a new dataset of 132k associated Perception sub-questions which humans perceive as containing the sub-questions needed to answer the original questions. Our dataset can be found at aka.ms/vqa-introspect. After validating the quality of the new dataset, we use it to perform fine-grained evaluation of state-of-the-art models, checking whether their reasoning is in line with their perception. We show that state-of-the-art VQA models have similar accuracy in answering perception and reasoning tasks but have problems with consistency; in 28.14% of the cases where models answer the reasoning question correctly, they fail to answer the corresponding perception sub-question, highlighting problems with consistency and the risk that models may be learning to answer reasoning questions through learning

common answers and biases.

Finally, we introduce SQuINT – a generic modeling approach that is inspired by the compositional learning paradigm observed in humans. SQuINT incorporates VQA-Introspect annotations into learning with a new loss function that encourages image regions important for the sub-questions to play a role in answering the main Reasoning questions. Empirical evaluations demonstrate that the approach results in models that are more consistent across Reasoning and associated Perception tasks with no loss of accuracy. We also find that SQuINT improves model attention maps for Reasoning questions, thus making models more trustworthy.

6.2 Related Work

Visual Question Answering [1], one of the most widely studied vision-and-language problems, requires associating image content with natural language questions and answers (thus combining perception, language understanding, background knowledge and reasoning). However, it is possible for models to do well on the task by exploiting language and dataset biases, e.g. answering "yellow" to "What color is the banana?" without regard for the image or by answering "yes" to most yes-no questions [**li-tell**, 7, 31, 32, 33]. This motivates additional forms of evaluation, e.g. checking if the model can understand question rephrasings [124] or whether it exhibits logical consistency [125]. In this work, we present a novel evaluation of questions that require reasoning capabilities, where we check for consistency between how models answer higher level Reasoning questions and how they answer corresponding Perception sub-questions.

A variety of datasets have been released with attention annotations on the image pointing to regions that are important to answer questions ([35, 36]), with corresponding work on enforcing such grounding [37, 38, 31]. Our work is complementary to these approaches, as we provide language-based grounding (rather than visual), and further evaluate the link between perception capabilities and how they are composed by models for answering Rea-

soning tasks. Closer to our work is the dataset of Lisa et al. [36], where natural language justifications are associated with (question, answer) pairs. However, most of the questions contemplated (like much of the VQA dataset) pertain to perception questions (e.g. for the question-answer "What is the person doing? Snowboarding", the justification is "...they are on a snowboard ..."). Furthermore, it is hard to use natural language justifications to evaluate models that do not generate similar rationales (i.e. most SOTA models), or even coming up with metrics for models that do. In contrast, our dataset and evaluation is in the same modality (QA) that models are already trained to handle.

6.3 Reasoning-VQA and VQA-Introspect

In the first part of this section, we present an analysis of the common type of questions in the VQA dataset and highlight the need for classifying them into Perception and Reasoning questions. We then define Perception and Reasoning questions and describe our method for constructing the Reasoning split. In the second part, we describe how we create the new VQA-Introspect dataset through collecting sub-questions and answers for questions in our Reasoning split. Finally, we describe experiments conducted in order to validate the quality of our collected data.

6.3.1 Perception vs. Reasoning

A common technique for finer-grained evaluation of VQA models is to group instances by answer type (yes/no, number, other) or by the first words of the question (what color, how many, etc) [1]. While useful, such slices are coarse and do not evaluate the model's capabilities at different points in the abstraction scale. For example, questions like "Is this a banana?" and "Is this a healthy food?" start with the same words and expect yes/no answers. While both test if the model can do object recognition, the latter requires additional capabilities in connecting recognition with prior knowledge about which food items are healthy and which are not. This is not to say that Reasoning questions are inherently

harder, but that they require both visual understanding and an additional set of skills (logic, prior knowledge, etc) while Perception questions deal mostly with visual understanding. For example, the question "How many round yellow objects are to the right of the smallest square object in the image?" requires very complicated visual understanding, and is arguably harder than "Is the banana ripe enough to eat?", which requires relatively simple visual understanding (color of the bananas) and knowledge about properties of ripe bananas. Regardless of difficulty, categorizing questions as Perception or Reasoning is useful for both detailed model evaluation based on capabilities and also improving learning, as we demonstrate in later sections. We now proceed to define these categories more formally.

Perception : We define Perception questions as those which can be answered by detecting and recognizing the existence, physical properties and / or spatial relationships between entities, recognizing text / symbols, simple activities and / or counting, and that do not require more than one hop of reasoning or general commonsense knowledge beyond what is visually present in the image. Some examples are: "Is that a cat? " (existence), "Is the ball shiny?" (physical property), "What is next to the table?" (spatial relationship), "What does the sign say?" (text / symbol recognition), "Are the people looking at the camera?" (simple activity), etc. We note that spatial relationship questions have been considered reasoning tasks in previous work [126] as they require lower-level perception tasks in composition to be answered. For our purposes it is useful to separate visual understanding from other types of reasoning and knowledge, and thus we classify such spatial relationships as Perception.

Reasoning : We define Reasoning questions as non-Perception questions which require the synthesis of perception with prior knowledge and / or reasoning in order to be answered. For instance, "Is this room finished or being built?", "At what time of the day would this meal be served?", "Does this water look fresh enough to drink?", "Is this a home or a hotel?", "Are the giraffes in their natural habitat?" are all Reasoning questions.

Our analysis of the perception questions in the VQA dataset revealed that most perception questions have distinct patterns that can be identified with high precision regex-based

rules. By handcrafting such rules (details can be found in [**squint˙arxiv**]) and filtering out perception questions, we identify 18% of the VQA dataset as highly likely to be Reasoning. To check the accuracy of our rules and validate their coverage of Reasoning questions, we designed a crowdsourcing task on Mechanical Turk that instructed workers to identify a given VQA question as Perception or Reasoning, and to subsequently provide sub-questions for the Reasoning questions, as described next. 94.7% of the times, trained workers classified our resulting questions as reasoning questions demonstrating the high precision of the regex-based rules we created.

6.3.2 VQA-Introspect data

Given the complexity of distinguishing between Perception / Reasoning and providing sub-questions for Reasoning questions, we first train and filter workers on Amazon Mechanical Turk (AMT) via qualification rounds before we rely on them to generate high-quality sub-questions.

Worker Training - We manually annotate 100 questions from the VQA dataset as Perception and 100 as Reasoning questions, to serve as examples. We first teach crowdworkers the difference between Perception and Reasoning questions by presenting definitions and showing several examples of each, along with explanations. Then, crowdworkers are shown (question, answer) pairs and are asked to identify if the given question is a Perception question or a Reasoning question [1]. Finally, for Reasoning questions, we ask workers to add all Perception questions and corresponding answers (in short) that would be necessary to answer the main question (details and interface can be found in [**squint˙arxiv**]). In this qualification HIT, workers have to make 6 Perception and Reasoning judgments, and they qualify if they get 5 or more answers right.

We launched further pilot experiments for the crowdworkers who passed the first qualification round, where we manually evaluated the quality of their sub-questions based on

[1] We also add an "Invalid" category to flag nonsensical questions or those which can be answered without looking at the image

whether they were Perception questions grounded in the image and sufficient to answer the main question. Among those 463 workers who passed the first qualification test, 91 were selected (via manual evaluation) as high-quality workers, which finally qualified for attempting our main task.

Main task - In the main data collection, all VQA questions identified as Reasoning by regex-rules and a random subset of the questions identified as Perception were further judged by workers (for validation purposes). We eliminated ambiguous questions by further filtering out questions where there is high worker disagreement about the answer. We required at least 8 out of 10 workers to agree with the majority answer for yes/no questions and 5 out of 10 for all other questions. This labeling step left us with a Reasoning split that corresponds to $\sim$13% of the VQA dataset.

At the next step. each <question, image> pair labeled as Reasoning had sub questions generated by 3 unique workers [2]. Removing duplicate question, answer pairs left on average 2.60 sub-questions per Reasoning question. Qualitative examples from the resulting dataset are presented in Fig. C.2.

The resulting VQA-Introspect v0.7 train, which contains sub questions for VQAv1 train, has 27441 Reasoning questions and the corresponding 79905 sub questions. The VQA-Introspect val has 15448 Reasoning questions (from whole VQAv2 val) and 52573 corresponding sub questions. This Reasoning split is not exhaustive, but is high precision (as demonstrated below) and contains questions that are not ambiguous, and thus is useful for evaluation and learning.

6.3.3 Dataset Quality Validation

In order to confirm that the sub-questions in VQA-Introspect are really Perception questions, we did a further round of evaluation with workers who passed the worker qualification task described in Section C.3 but had not provided sub-questions for our main task.

[2] A small number of workers displayed degraded performance after the qualification round, and were manually filtered

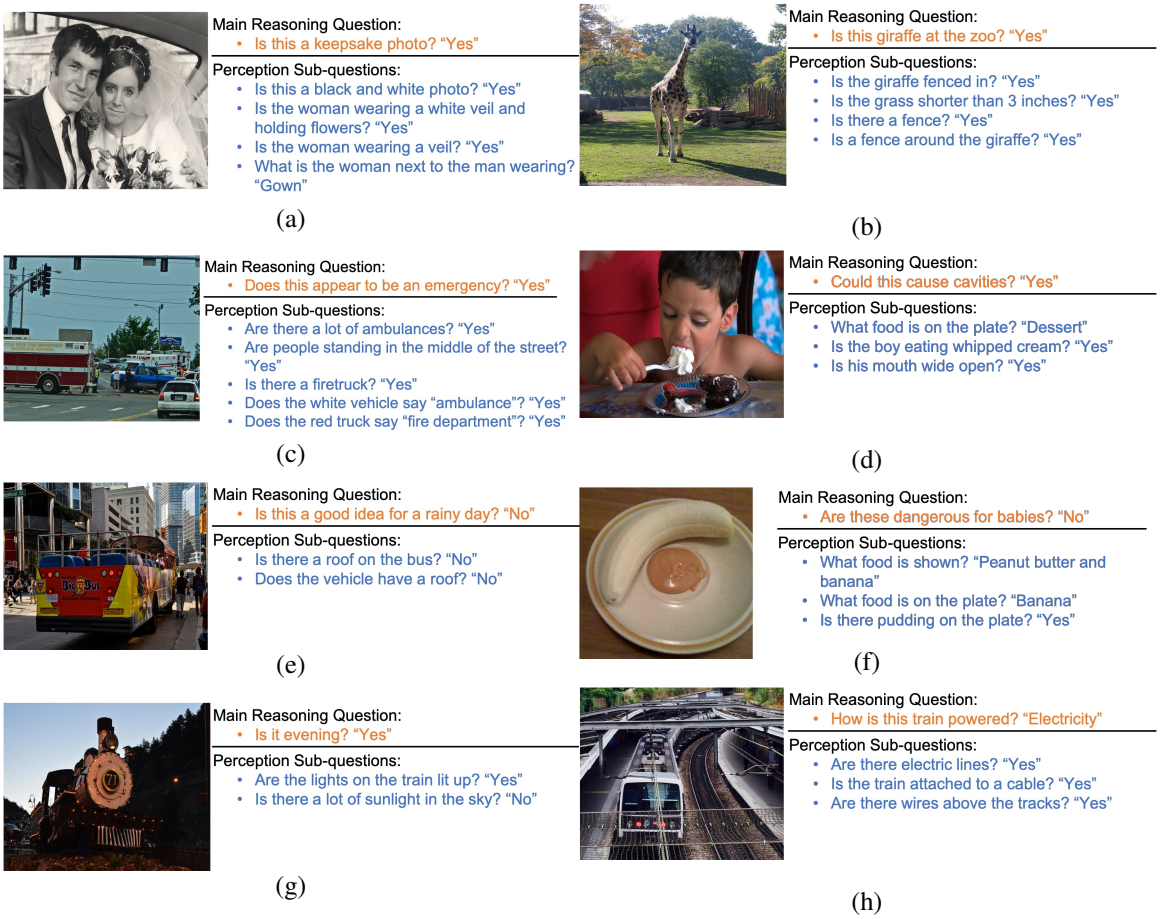


Figure 6.2: Qualitative examples of Perception sub-questions in our VQA-Introspect dataset for main questions in the Reasoning split of VQA. Main questions are in orange and sub questions are in blue. A single worker may have provided more than one sub questions for the same (image, main question) pair.

In this round, 87.8% of sub-questions in VQA-Introspect were judged to be Perception questions by at least 2 out of 3 workers.

It is crucial for the semantics of VQA-Introspect that the sub-questions are tied to the original Reasoning question. While verifying that the sub-questions are necessary to answer the original question requires workers to think of all possible ways the original question could be answered (and is thus too hard), we devised an experiment to check if the sub-questions provide at least sufficient visual understanding to answer the Reasoning question. In this experiment, workers are shown the sub-questions with answers, and then asked to answer the Reasoning question without seeing the image, thus having to rely only on the visual knowledge conveyed by the sub-questions. At least 2 out of 3 workers were able to

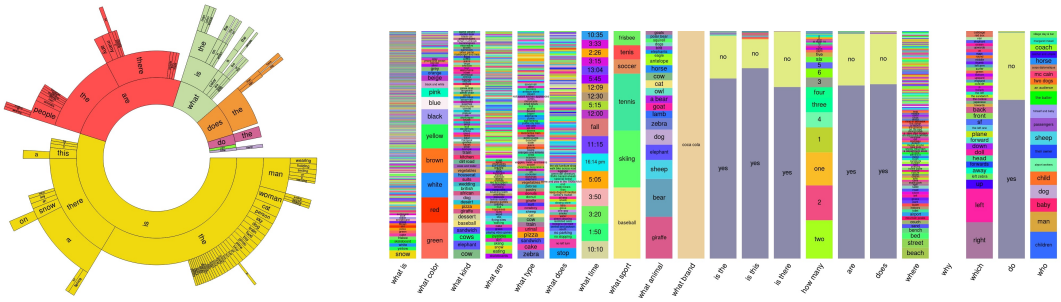


Figure 6.3: Left: Distribution of questions by their first four words. The arc length is proportional to the number of questions containing the word. White areas are words with contributions too small to show, Right: Distribution of answers per question type

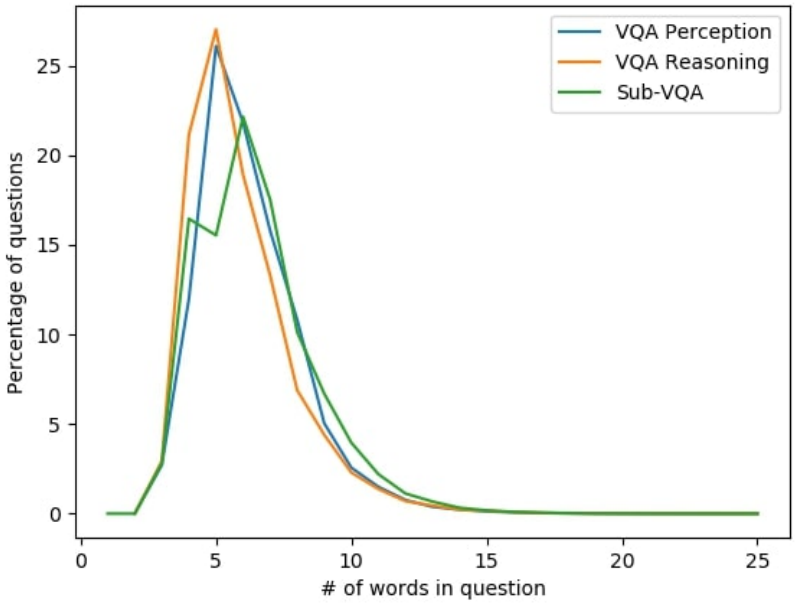


Figure 6.4: Percentage of questions with different word lengths for the train and val sub-questions of our Sub-VQA dataset.

answer 89.3% of the Reasoning questions correctly in this regime (95.4% of binary Reasoning questions). For comparison, when we asked workers to answer Reasoning questions with no visual knowledge at all (no image and no sub-questions), this accuracy was 52% (58% for binary questions). These experiments give us confidence that the sub-questions in VQA-Introspect are indeed Perception questions that convey components of visual knowledge which can be composed to answer the original Reasoning questions.

6.4 Dataset Analysis

The distribution of questions in our VQA-Introspect dataset is shown in Figure 6.3. It is interesting to note that comparing these plots with those for the VQA dataset [1] show that the VQA-Introspect dataset questions are more specific. For example, there are 0 "why" questions in the dataset which tend to be reasoning questions. Also, for "where" questions, a very common answer in VQA was "outside" but answers are more specific in our VQA-Introspect dataset (e.g., "beach", "street"). Figure 6.4 shows the distribution of question lengths in the Perception and Reasoning splits of VQA and in our VQA-Introspect dataset. We see that most questions range from 4 to 10 words. Lengths of questions in the Perception and Reasoning splits are quite similar, although questions in VQA-Introspect are slightly longer (the curve is slightly shifted to the right), possibly on account of the increase in specificity/detail of the questions.

One interesting question is whether the main question and the sub-questions deal with the same concepts. In order to explore this, we used noun chunks surrogates for concepts [3], and measured how often there was any overlap in concepts between the main question and the associated sub-question. Noun-chunks are only a surrogate and may miss semantic overlap otherwise present (e.g. through verb-noun connections like "fenced" and "a fence" in Figure C.2 (b), sub-questions). With this caveat, we observe that there is overlap only 19.19% of the time, indicating that Reasoning questions in our split often require knowledge about concepts not explicitly mentioned in the corresponding Perception questions. The lack of overlap indicates that models cannot solely rely on visual perception in answering Reasoning tasks, but incorporating background knowledge and common sense understanding is necessary. For example, in the question "Is the airplane taking off or landing?", the concepts present are 'airplane' and 'landing', while for the associated sub-question "Are the wheels out?", the concept is 'wheels'. Though 'wheels' do not occur in the main question, the concept is important, in that providing this grounding might help the model

[3] Concepts are extracted with the Python spaCy library.

explicitly associate the connection between airplane wheels and take-offs / landings.

6.5 Fine grained evaluation of VQA Reasoning

VQA-Introspect enables a more detailed evaluation of the performance of current state-of-the-art models on Reasoning questions by checking whether correctness on these questions is consistent with correctness on the associated Perception sub-questions. It is important to notice that a Perception failure (an incorrect answer to a sub-question) may be due to a problem in the vision part of the model or a grounding problem – the model in Figure 6.5 may know that the banana is mostly yellow and use that information to answer the ripeness question, while, at the same time, fail to associate this knowledge with the word "yellow", or fail to understand what the sub-question is asking. While grounding problems are not strictly visual perception failures, we still consider them Perception failures because the goal of VQA is to answer natural language questions about an image, and the sub-question being considered pertain to Perception knowledge as defined previously. With this caveat, there are four possible outcomes when evaluating Reasoning questions with associated Perception sub-questions, which we divide into four quadrants:

Q1: Both main & sub-questions correct (M✓ S✓): While we cannot claim that the model predicts the main question correctly *because* of the sub-questions (e.g. the bananas are ripe *because* they are mostly yellow), the fact that it answers both correctly is consistent with good reasoning, and should give us more confidence in the original prediction.

Q2: Main correct & sub-question incorrect (M✓ S✗): The Perception failure indicates that there might be a reasoning failure. While it is possible that the model is composing other perception knowledge that was not captured by the identified sub-questions (e.g. the bananas are ripe because they have black spots on them), it is also possible (and more likely) that the model is using a spurious shortcut or was correct by random chance.

Q3: Main incorrect & sub-question correct (M✗ S✓): The Perception failure here indicates a clear reasoning failure, as we validated that the sub-questions are sufficient to

answer the main question. In this case, the model knows that the bananas are mostly yellow and still thinks they are not ripe enough, and thus it failed to make the "yellow bananas are ripe" connection.

Q4: Both main & sub-question incorrect (M✗ S✗): While the model may not have the reasoning capabilities to answer questions in this quadrant, the Perception failure could explain the incorrect prediction.

In sum, Q2 and Q4 are definitely Perception failures, Q2 likely contains Reasoning failures, Q3 contains Reasoning failures, and we cannot judge Reasoning in Q4.

As an example, we evaluate the Pythia model [127] (SOTA as of 2018)[4] along these quadrants (Table 6.1) for the Reasoning split of VQA. The overall accuracy of the model is 60.26%, while accuracy on Reasoning questions is 65.99%. We note that for 28.14% of the cases, the model is inconsistent, i.e., it answered the main question correctly, but got the sub question wrong. Further, we observe that 14.92% of the times the Pythia model gets *all* the sub questions wrong when the main question is right – *i.e.*, it seems to be severely wrong on its perception and using other paths (shortcuts or biases) to get the Reasoning question right .

6.6 Improving learned models with VQA-Introspect

In this section, we consider how VQA-Introspect can be used to improve models that were trained on VQA datasets. Our goal is to reduce the number of possible reasoning or perception failures (M✓ S✗ and M✗ S✓) without diminishing the original accuracy of the model.

6.6.1 Finetuning

The simplest way to incorporate VQA-Introspect into a learned model is to fine-tune the model on it. However, a few precautions are necessary: we make sure that sub-questions

[4]source: `https://visualqa.org/roe_2018.html`

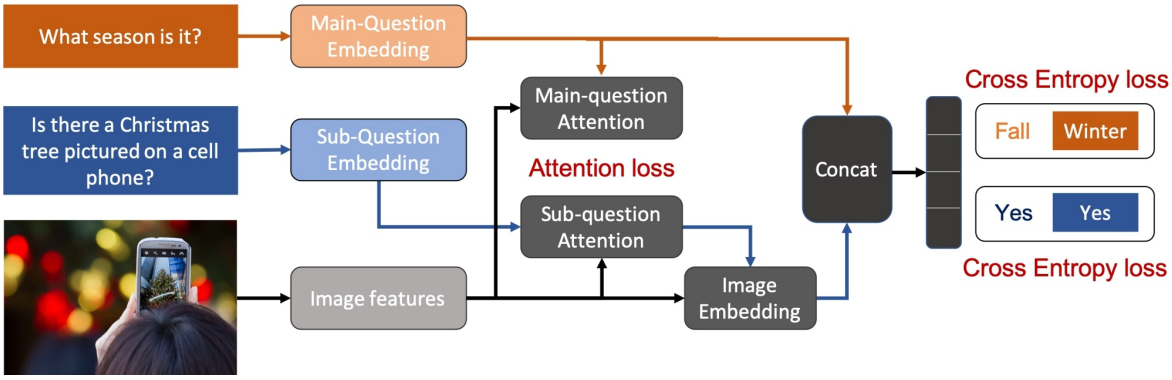


Figure 6.5: Sub-Question Importance-aware Network Tuning (SQuINT) approach: Given an image, a Reasoning question like "What season is it?" and an associated Perception sub-question like "Is there a Christmas tree pictured on a cell phone?", we pass them through the Pythia architecture [127]. The loss function customized for SQuINT is composed of three components: an attention loss that penalizes for the mismatch between attention for the main-question and the attention for the sub-question based on an image embedding conditioned on sub-question and image features, a cross entropy loss for answer of the main-question and a cross entropy loss for the answer of the sub-question. The loss function encourages the model to get the answers of both the main-question and sub-question right simultaneously, while also encouraging the model to use the right attention regions for the reasoning task.

always appear on the same batch as the original question, and use the averaged binary cross entropy loss for the main question and the sub question as a loss function. Furthermore, to avoid catastrophic forgetting [128] of the original VQA data during finetuning, we augment every batch with randomly sampled data from the original VQA dataset. In our empirical evaluations, we compare this approach with fine-tuning on the same amount of randomly sampled Perception questions from VQAv2.

6.6.2 Sub-Question Importance-aware Network Tuning (SQuINT)

The intuition behind Sub-Question Importance-aware Network Tuning (SQuINT) is that a model should attend to the same regions in the image when answering the Reasoning questions as it attends to when answering the associated Perception sub-questions, since they capture the visual components required to answer the main question. SQuINT does this by learning how to attend to sub-question regions of interest and reasoning over them to answer the main question. We now describe how to construct a loss function that captures this intuition.

Attention loss - As described in Section 6.3, the sub-questions in the dataset are simple perception questions asking about well-grounded objects/entities in the image. Current well-performing models based on attention are generally good at visually grounding regions in the image when asked about simple Perception questions, given that they are trained on VQA datasets which contain large amounts of Perception questions. In order to make the model look at the associated sub-question regions while answering the main question, we apply a Mean Squared Error (MSE) loss over the the spatial and bounding box attention weights.

Cross Entropy loss - While the attention loss encourages the model to look at the right regions given a complex Reasoning question, we need a loss that helps the model learn to reason given the right regions. Hence we apply the regular Binary Cross Entropy loss on top of the answer predicted for the Reasoning question given the sub-question attention. In addition we also use the Binary Cross Entropy loss between the predicted and GT answer for the sub-question.

Total SQuINT loss - We jointly train with the attention and cross entropy losses. Let A_{reas} and A_{sub} be the model attention for the main reasoning question and the associated sub-question, and gt_{reas} and gt_{sub} be the ground-truth answers for the main and sub-question respectively. Let $o_{reas}|A_{sub}$ be the predicted answer for the reasoning question given the attention for the sub-question. The SQuINT loss is formally defined as:

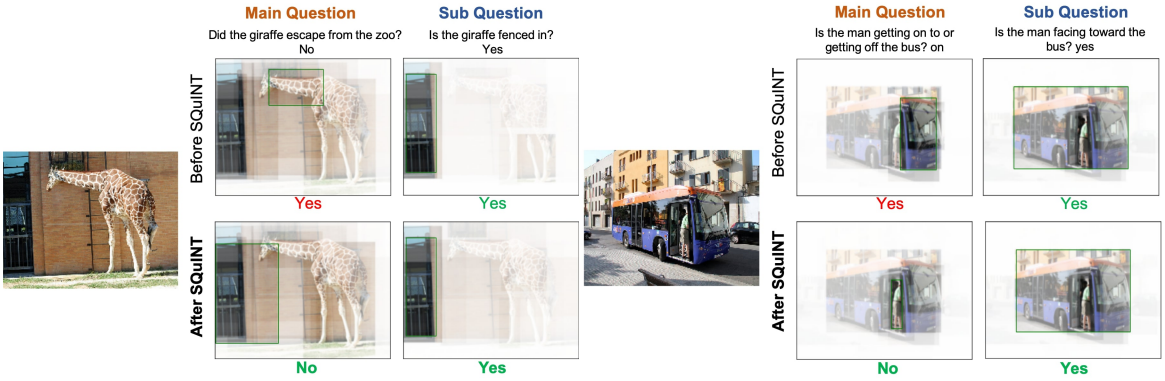
$$\begin{aligned}\mathcal{L}_{\textbf{SQuINT}} = \text{MSE}(A_{reas}, A_{sub}) \\ + \text{BCE}(o_{reas}|A_{sub}, gt_{reas}) + \text{BCE}(o_{sub}, gt_{sub})\end{aligned}$$

The first term encourages the network to look at the same regions for reasoning and associated perception questions, while the second and third terms encourage the model to give the right answers to the questions given the attention regions. The loss is simple and can be applied as a modification to any model that uses attention.

Table 6.1: Results on held out VQAv2 validation set for (1) Consistency metrics along the four quadrants described in Section 6.5 and Consistency and Attention Correlation metrics as described in Section 6.5 (metrics), and (2) Overall and Reasoning accuracy. The Reasoning accuracy is obtained by only looking at the number of times the main question is correct (M✓ S✓ + M✓ S✗) .

Method	Consistency Metric							VQA Accuracy	
	M✓ S✓ ↑	M✓ S✗ ↓	M✗ S✓ ↓	M✗ S✗ ↓	Consistency% ↑	Consistency% (balanced) ↑	Attn Corr ↑	Overall ↑	Reasoning (M✓ S✓ + M✓ S✗) ↑
Pythia	47.42	18.57	20.70	13.31	71.86	69.57	0.71	60.26	65.99
Pythia + VQA-Introspect data	52.54	13.55	22.50	11.41	**79.50**	**75.44**	0.71	60.20	66.09
Pythia + VQA-Introspect + SQuINT	**52.56**	13.84	22.38	11.22	79.25	75.26	**0.74**	**60.33**	**66.41**

6.7 Experiments



(a) (b)

Figure 6.6: Qualitative examples showing the model attention before and after applying SQuINT. (a) shows an image along with the reasoning question, *'Did the giraffe escape from the zoo?'*, for which the Pythia model looks at somewhat irrelevant regions and answers "Yes" incorrectly. Note how the same model correctly looks at the fence to answer the easier sub-question, *'Is the giraffe fenced in?'*. After applying SQuINT, which encourages the model to use the perception based sub question attention while answering the reasoning question, it now looks at the fence and correctly answers the main reasoning question.

In this section, we perform fine grained evaluation of VQA reasoning as detailed in Section 6.5, using the SOTA model **Pythia** [127] as a base model (although any model that uses visual attention would suffice). We trained the base model on VQAv1, and evaluated the baseline and all variants on the Reasoning split and corresponding VQA-Introspect sub-questions of VQAv2. As detailed in Section 6.6, **Pythia + VQA-Introspect data** corresponds to finetuning the base model on train VQA-Introspect v0.7 subquestions *of VQAv1*, while **Pythia + VQA-Introspect + SQuINT** finetunes Pythia + VQA-Introspect such that it now attends to the same regions for main questions and associated sub-questions

(again, of VQA-Introspect v0.7). For direct comparisons with Pythia + VQA-Introspect + SQuINT, during Pythia + VQA-Introspect finetuning, we added both the main question and sub-question in the same batch. In Table 6.1, we report the reasoning breakdown detailed in Section 6.5. We also report a few additional metrics: **Consistency** refers to how often the model predicts the sub-question correctly given that it answered the main question correctly, while **Consistency (balanced)** reports the same metric on a balanced version of the sub-questions (to make sure models are not exploiting biases to gain consistency). **Attention Correlation** refers to the correlation between the attention embeddings of the main and sub-question. Finally, we report **Overall** accuracy (on the whole evaluation dataset), and accuracy on the Reasoning split (**Reasoning Accuracy**). Note that our approach does not require sub-questions at test time.

The results in Table 6.1 indicate that fine-tuning on VQA-Introspect (using data augmentation or SQuINT), increases consistency without hurting accuracy or Reasoning accuracy. Correspondingly, our confidence that it actually learned the necessary concepts when it answered Reasoning questions correctly should increase.

The **Attention Correlation** numbers indicate that SQuINT really is helping the model use the appropriate visual grounding (same for main-question as sub-questions) at test time, even though the model was trained on VQAv1 and evaluated on VQAv2. This effect does not seem to happen with naive finetuning on VQA-Introspect. We present qualitative validation examples in Figure 6.6, where the base model attends to irrelevant regions when answering the main question (even though it answers correctly), while attending to relevant regions when asked the sub-question. The model finetuned on SQuINT, on the other hand, attends to regions that are actually informative in both main and sub-questions (notice that this is evaluation, and thus the model is not aware of the sub-question when answering the main question and vice versa). This is further indication that SQuINT is helping the model reason in ways that will generalize when it answers Reasoning questions correctly, rather than use shortcuts. One other way to show the benefit of relevant sub-questions (from

VQA-Introspect) on improving reasoning accuracy could be by comparing the effects of aligning main question attention with attention of relevant sub-questions as opposed to aligning main question attention with attention of a sub-question that is irrelevant to answering the main reasoning question.

6.8 Discussion and Future Work

The VQA task requires multiple capabilities in different modalities and at different levels of abstraction. We introduced a hard distinction between Perception and Reasoning which we acknowledge is a simplification of a continuous and complex reality, albeit a useful one. In particular, linking the perception components that are needed (in addition to other forms of reasoning) to answer reasoning questions opens up an array of possibilities for future work, in addition to improving evaluation of current work. We proposed preliminary approaches that seem promising: fine-tuning on VQA-Introspect and SQuINT both improve the consistency of the SOTA model with no discernible loss in accuracy, and SQuINT results in qualitatively better attention maps. We expect future work to use VQA-Introspect even more explicitly in the modeling approach, similar to current work in explicitly composing visual knowledge to improve *visual* reasoning [129]. In addition, similar efforts to ours could be employed at different points in the abstraction scale, e.g. further dividing complex Perception questions into simpler components, or further dividing the Reasoning part into different forms of background knowledge, logic, etc. We consider such efforts crucial in the quest to evaluate and train models that truly generalize, and hope VQA-Introspect spurs more research in that direction.

CHAPTER 7

DISCUSSION

There exists several open challenges in the science of explainability. Firstly, explainability/explanations currently is an overloaded term with different papers using the term differently. While it is good to have a commonly agreed definition, it is at least important to define the term upfront in the specific context of the work. Secondly, it is important to understand the value explanations add to the task/model – whether the task really needs explanations or is it just something good to have. This would help developers provide explanations that lie in the right place in the interpretability vs faithfulness spectrum Sec. 3.4.3. Thirdly, explanations have to be well served to specific target audience. An explanation catered for the doctor to gain trust in the system's prediction need not necessarily help the developer in improving the model. Fourthly, evaluation of explanations should also factor in the time-budget, expertise and biases of different human evaluators. An explanation that is based on hundreds of parameters is neither easily understandable nor is useful to the end user with limited time-budget. The expertise of the user tends to play a big role in whether they would be able to understand the given explanation. Humans have a tendency to oversee non-existing patterns in visual data – they tend to show confirmation bias. One way to overcome this is by complementing human evaluators with carefully constructed automated metrics. Finally, in order to improve models by providing feedback on the explanations, the need for human domain experts is crucial. To conclude, it is important to understand the limitations of explanations and calibrate our expectations from explanation techniques so as to avoid unpleasant surprises.

CHAPTER 8
CONCLUSION

In this thesis, we first introduced a technique for explaining decisions from a wide variety of Convolutional Neural Network (CNN) based deep networks, called Gradient-weighted Class Activation Mapping (Grad-CAM). Grad-CAM uses the gradients of any target concept (say 'dog' in a classification network or a sequence of words in captioning network) flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. The invention of Grad-CAM and our analysis revealed a number of interesting and novel findings – We found that even simple non-attention based captioning and VQA models learn to look at relevant image regions when making decisions. We showed how explanations can not only help establish trust with humans, but also help untrained users successfully discern a stronger network from a weaker one, *even when both make identical predictions.* We also made a first attempt at showing how explanations help in diagnosing failure modes of current deep models, and in uncovering biases in datasets.

Following that, we came up with several ways of using Grad-CAM for improving training. Towards that, we showed how Grad-CAM can help incorporate domain knowledge into deep networks in order to learn novel concepts. We introduced a zero-shot learning (ZSL) approach based on mapping unseen class descriptions to Grad-CAM neuron importance within a deep network and then optimizing unseen classifier weights to effectively combine these concepts. In contrast to previous ZSL approaches, our method is capable of explaining predictions with human-interpretable semantics.

When explanation modalities such as Grad-CAM are employed to assess the evidence that current vision and language models are basing their decisions on, we find that they are often relying on spurious correlations in the training data. To address this, we explore if

giving a *small hint* in the form of human attention can help improve grounding and reliability. we introduced Human Importance-aware Network Tuning (HINT), which enforces a ranking loss between human annotations of input importance and Grad-CAM explanations from a deep network – updating model parameters via a gradient-of-gradient step. Importantly, this constrains models to not only look at the correct regions but to also be sensitive to the content present there when making predictions. We apply HINT to two tasks – Visual Question Answering (VQA) and image captioning – and find our approach that forces visual grounding also significantly improves task performance and human trustworthiness. While we experiment with HINT in the context of vision-and-language problems, the approach itself is general and can be applied to focus model decisions on specific inputs in any context. This is another example showing how Grad-CAM explanations can help improve models.

We recognized that answering complex reasoning-based questions in the VQA dataset requires more than just looking at the right regions. For such questions, we to also check and ensure that models learn to use the right reasoning on top of these regions. In chapter 6 we analysed the *reasoning abilities* of current Visual Question Answering (VQA) models. We noticed that current VQA models have consistency issues – they are able to answer seemingly harder reasoning questions right (e.g. "Is the banana ripe enough to eat?"), but fail on simpler, perception questions (e.g., "Are the bananas mostly green or yellow?") – indicating that the model possibly answered the original question for the wrong reasons, even if the answer was right. In order to quantify the extent to which this phenomenon occurs, we collected a new dataset of *perception* sub-questions for questions in the VQA dataset requiring reasoning abilities and observed that state-of-the-art models are inconsistent $\sim$30% of the time. We then proposed an approach which encourages the model to attend to the same parts of the image when answering the reasoning question and the perception sub-questions. The key takeaways of our work are: 1) it is important to use trust metrics (such as consistency & reliability) besides accuracy, and 2) it is important to

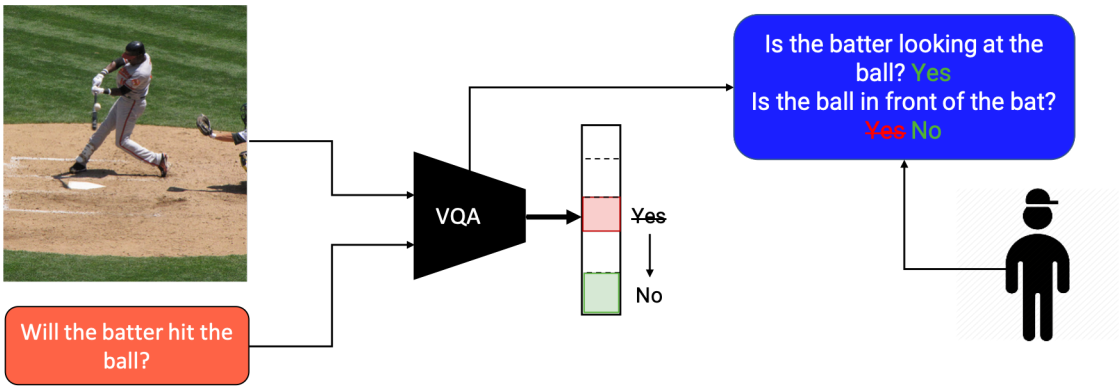


Figure 8.1: Example showing how supervising natural language explanations from VQA models can help us fix them.

understand our data well and develop models to use it efficiently, i.e., lesser data collected and used in a specific way can be more useful than large amounts of data collected and used in a generic way. This leads to better reasoning models which cannot easily rely on undesirable shortcuts, thereby making models *right for right reasons*.

To summarize, in the first part we introduced a technique, Grad-CAM, to visually explain model decisions from a wide variety of architectures and in the following works we used Grad-CAM explanations to improve specific parts of models.

8.1 Future work directions

8.1.1 Modality-specific Explanations

Even for tasks involving multiple modalities such as VQA, we primarily focused on explaining decisions in the visual space. The multi-modal task of VQA has a language component which has not been explored in the context of explanations before. Natural language explanations can offer much more deeper understanding giving us information about how models reason on top on important regions used by them in the decision making process. We could use the VQA-introspect dataset from Chapter 6 to see which sub-questions the model relies on when answering a reasoning question.

How can natural-language explanations help? We see that sub-question based explanations can help improve models in 2 scenarios. First, if the predicted question explanations

does not match any of the sub-questions (from VQA-introspect), it would indicate that the model does not know the right set of sub-question that it needs to answer before answering the main reasoning question. We could provide supervision and enforce models to use the correct sub-question so models rely on the right perceptual concepts when answering questions. Secondly, if the model relies on the right sub-question but answers the sub-question incorrectly, we could see if forcing the model to correctly answer the associated sub-question would help the model answer the reasoning question correctly. As an illustration, see Fig 8.1 where the model initially incorrectly answers the reasoning question, "Will the batter hit the ball?". Through the sub-question explanation we can see that the model uses the right perceptual questions, "Is the batter looking at the ball?", and "Is the ball in front of the bat?". However it seems to answer the latter question incorrectly, indicating that it is likely that the model thinks that the ball is in front of the bat and hence thought that the batter will hit the ball. If we fix the answer to the sub-question, we can expect the model to change the answer to the main-question, "Will the batter hit the ball?" from 'yes' to 'no'. So this can provide an intuitive way to fix model decisions.

8.1.2 Explaining decisions from temporal models

Some examples of temporal tasks are video classification, time-series prediction, or vision-based navigation. These tasks require reasoning over inputs at different points of time. RNNs/LSTMs are typically used to capture the temporal structure required for these tasks. However we do not yet have a clear understanding of what hidden states in an RNN/LSTM learn. Understanding what individual neurons in the hidden states learn to capture at different times can help us better understand how models capture the temporal context required for performing the task. Similar to HINT, we can use such temporal explanations to provide feedback to encourage models to refer to relevant information in the past or make them remember important concepts.

8.1.3 Incorporating domain knowledge/rules into deep networks

Paired data (input with corresponding labels) are often an indirect way to teach AI. They are often not sufficient to capture domain knowledge. In most cases it is not possible to capture all the nuances and possible specifications associated with domain knowledge with just paired data. Like we have seen before, models tend to use undesirable shortcuts when learning mappings between inputs and labels. What this indicates is that specifying the exact goal for an ML model is a hard problem. We have seen that explanations help us understand what models learn. In order to close the gap between the learned function and the actual underlying function it is clear that explanations can play a huge role.

Domain knowledge is not always easily specified in the form of paired data. Sometimes, it is easy for us to provide this feedback in a form most natural to us (e.g., language). We would eventually want to move to agents which understand this form of feedback (which is obvious to us) and use it in interpretable ways to correct itself for when it comes across similar instances in the future. For example, we can explore ways of fixing a biased Doctor vs Nurse classifier by simply providing a natural language feedback such as, "For predicting 'nurse' do not focus on the 'gender' of the person."

In chapter 4, we explored how interpretability can serve as a medium to incorporate human domain knowledge in the form of natural language to extend a classifier to detect new classes. This is a preliminary step towards incorporating domain knowledge through a different modality. There is scope to extend this idea to be much more general.

Appendices

APPENDIX A

APPENDIX FOR GRAD-CAM

A.1 Appendix Overview

In the appendix, we provide:

I - More qualitative examples for image captioning and VQA.

II - More qualitative results for the bias experiment

III - More weakly-supervised segmentation results

IV - More details of Pointing Game evaluation technique

V - Qualitative comparison to existing visualization techniques

VI - More qualitative examples of textual explanations

VII - Grad-CAM for ResNet architectures

A.2 Qualitative results for vision and language tasks

In this section we provide more qualitative results for Grad-CAM and Guided Grad-CAM applied to the task of image captioning and VQA.

1. Image Captioning

We use the publicly available Neuraltalk2 code and model[1] for our image captioning experiments. The model uses VGG-16 to encode the image. The image representation is passed as input at the first time step to an LSTM that generates a caption for the image. The model is trained end-to-end along with CNN finetuning using the COCO [24] Captioning dataset. We feedforward the image to the image captioning model to obtain a caption. We use Grad-CAM to get a coarse localization and combine it with Guided Backpropagation to get

[1] `https://github.com/karpathy/neuraltalk2`

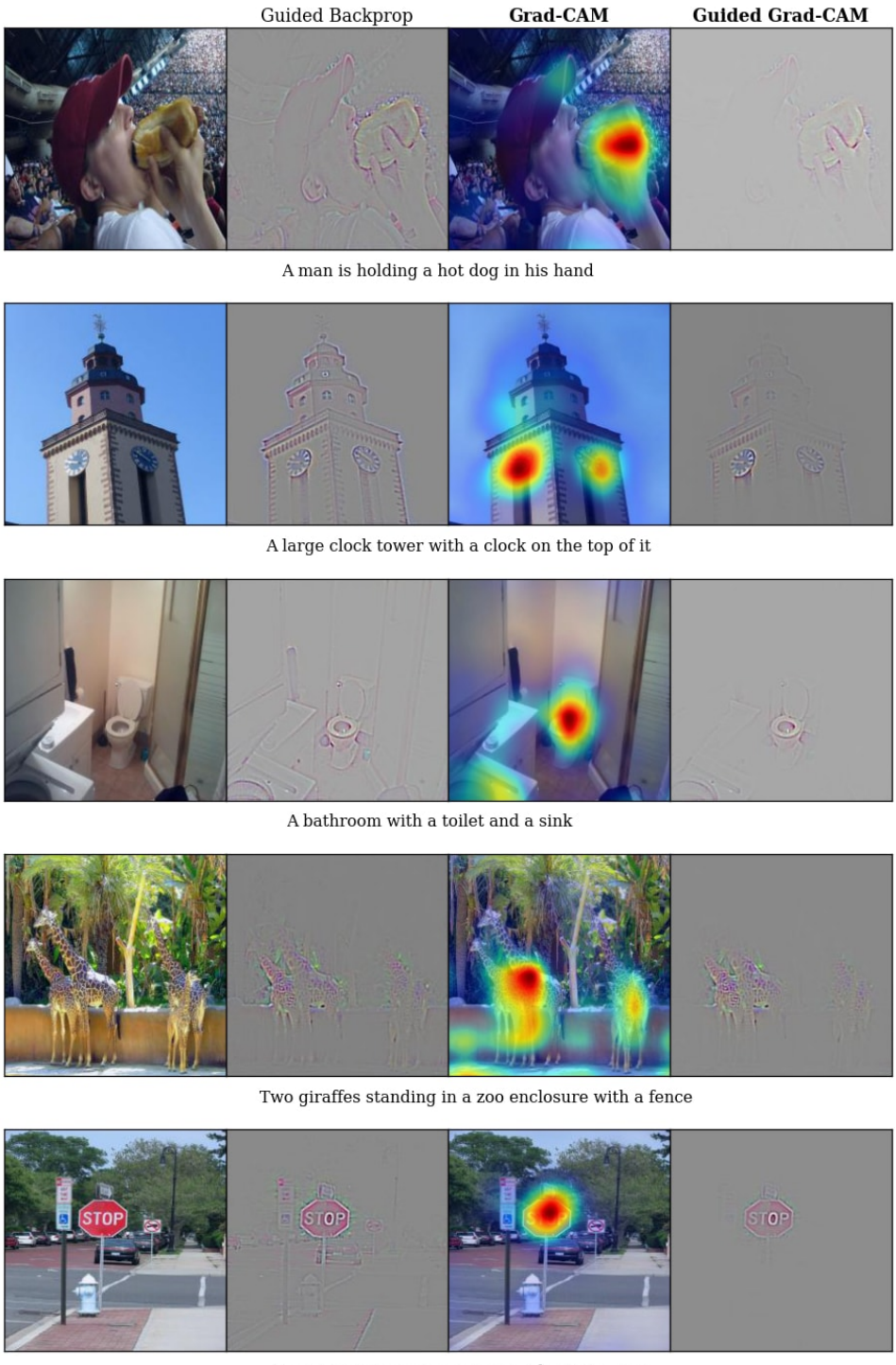


Figure A.1: Guided Backpropagation, Grad-CAM and Guided Grad-CAM visualizations for the captions produced by the Neuraltalk2 image captioning model.

a high-resolution visualization that highlights regions in the image that provide support for the generated caption.

A.3 Identifying and removing bias in datasets

In this section we provide qualitative examples showing the explanations from the two models trained for distinguishing doctors from nurses- model1 which was trained on images (with an inherent bias) from a popular search engine, and model2 which was trained on a more balanced set of images from the same search engine.

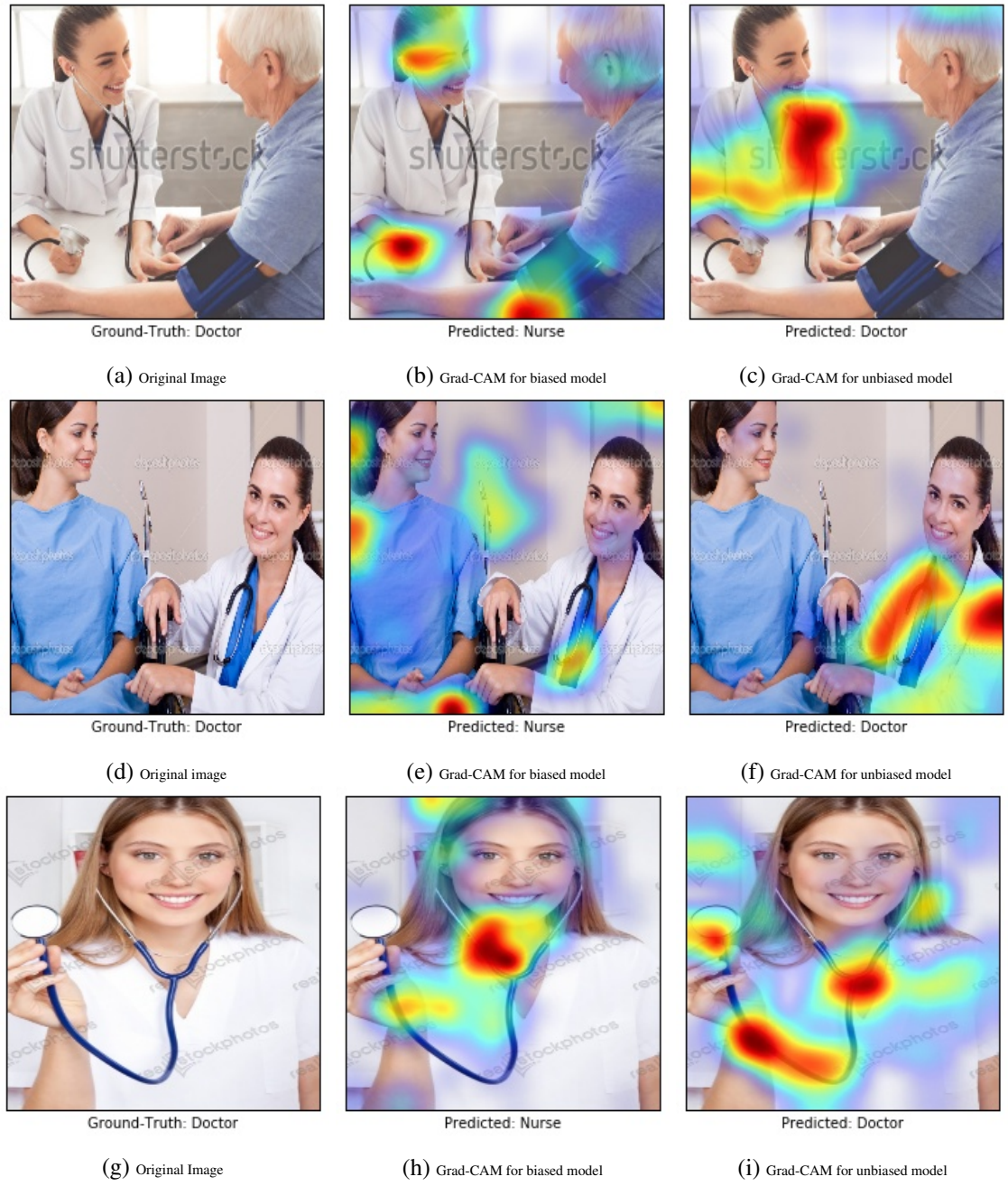
As shown in Fig. A.2, Grad-CAM visualizations of the model (model1) predictions show that the model had learned to look at the person's face / hairstyle to distinguish nurses from doctors, thus learning a gender stereotype.

Using the insights gained from the Grad-CAM visualizations, we balanced the dataset and retrained the model. The new model, model2 not only generalizes well to a balanced test set, it also looks at the right regions, as can be seen in Fig. A.2.

A.4 Weakly-supervised segmentation

In this section we provide more qualitative examples for weakly-supervised segmentation using Grad-CAM as seed for SEC ([64]).

The last row shows 2 failure cases. In the bottom left image, the clothes of the 2 person weren't highlighted correctly. This could be because the most discriminative parts are their faces, and hence Grad-CAM maps only highlights those. This results in a segmentation that only highlights the faces of the 2 people. In the bottom right image, the bicycles, being extremely thin aren't highlighed. This could be because the resolution of the Grad-CAM maps are low (14×14) which makes it difficult to capture thin areas.



(a) Original Image (b) Grad-CAM for biased model (c) Grad-CAM for unbiased model

(d) Original image (e) Grad-CAM for biased model (f) Grad-CAM for unbiased model

(g) Original Image (h) Grad-CAM for biased model (i) Grad-CAM for unbiased model

Figure A.2: Grad-CAM explanations for model1 and model2. In all the 3 examples we see that the biased model was looking at the face of the person to predict 'nurse' incorrectly, whereas the unbiased model looks at the stethoscope and the white coat to correctly predict 'doctor'.

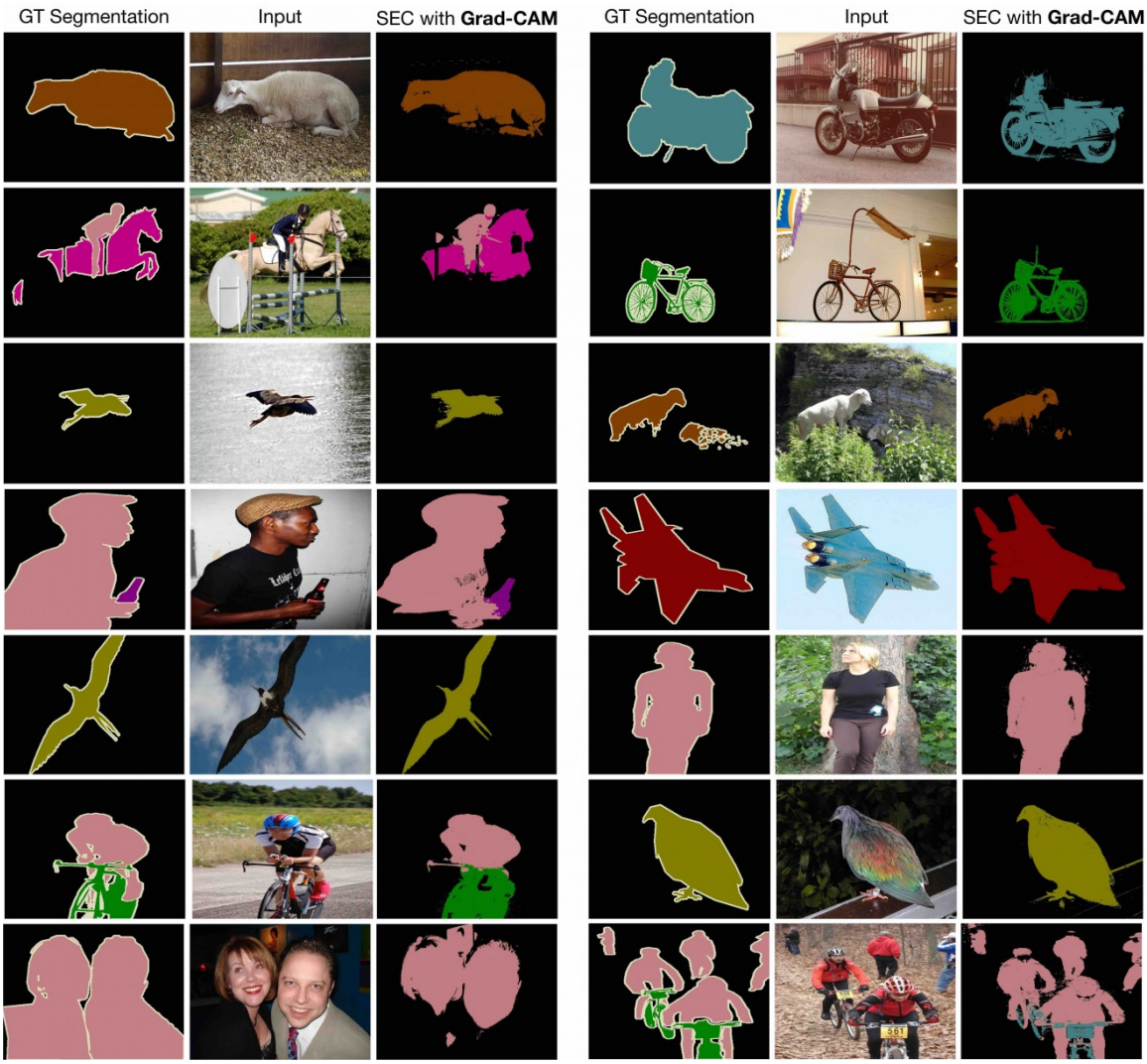


Figure A.3: PASCAL VOC 2012 Segmentation results with Grad-CAM as seed for SEC [64].

A.5 More details of Pointing Game

In [63], the pointing game was setup to evaluate the discriminativeness of different attention maps for localizing ground-truth categories. In a sense, this evaluates the precision of a visualization, *i.e*. how often does the attention map intersect the segmentation map of the ground-truth category. This does not evaluate how often the visualization technique produces maps which do not correspond to the category of interest.

Hence we propose a modification to the pointing game to evaluate visualizations of the top-5 predicted category. In this case the visualizations are given an additional option to reject any of the top-5 predictions from the CNN classifiers. For each of the two visualizations, Grad-CAM and c-MWP, we choose a threshold on the max value of the visualization, that can be used to determine if the category being visualized exists in the image.

We compute the maps for the top-5 categories, and based on the maximum value in the map, we try to classify if the map is of the GT label or a category that is absent in the image. As mentioned in Chapter 3, we find that our approach Grad-CAM outperforms c-MWP by a significant margin (70.58% vs 60.30% on VGG-16).

A.6 Qualitative comparison to Excitation Backprop (c-MWP) and CAM

In this section we provide more qualitative results comparing Grad-CAM with CAM [17] and c-MWP [63] on Pascal [130].

We compare Grad-CAM, CAM and c-MWP visualizations from ImageNet trained VGG-16 models finetuned on PASCAL VOC 2012 dataset. While Grad-CAM and c-MWP visualizations can be directly obtained from existing models, CAM requires an architectural change, and requires re-training, which leads to loss in accuracy. Also, unlike Grad-CAM, c-MWP and CAM can only be applied for image classification networks. Visualizations for the ground-truth categories can be found in Fig. A.4. ualitative examples comparing Grad-CAM with existing approaches can be found in [65].

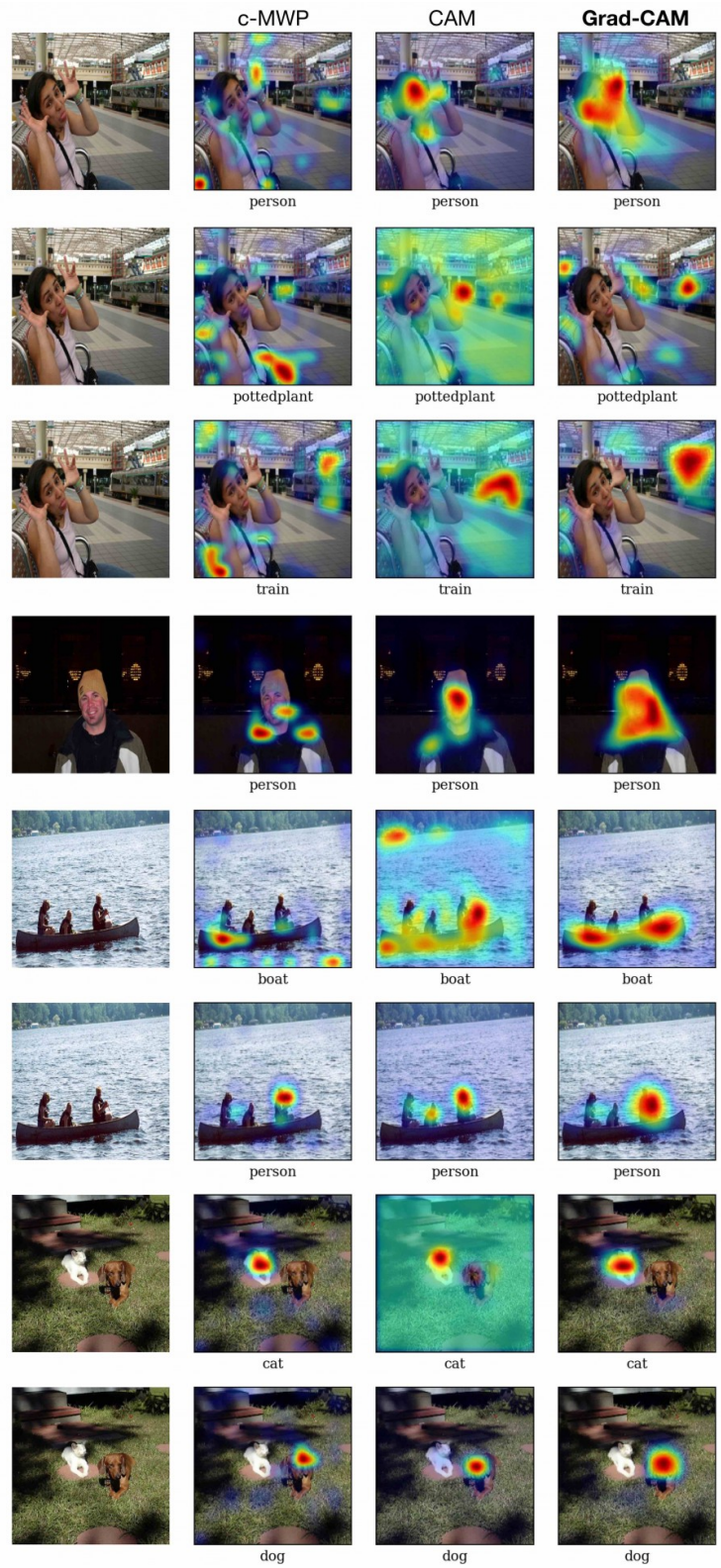


Figure A.4: Visualizations for ground-truth categories (shown below each image) for images sampled from the PASCAL [130] validation set.

A.7 Visual and Textual explanations for Places dataset

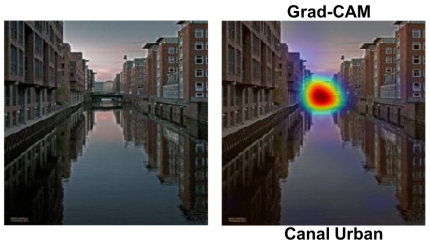
Fig. A.5 shows more examples of visual and textual explanations (Sec. 3.6) for the image classification model (VGG-16) trained on Places365 dataset ([68]).

A.8 Analyzing Residual Networks

In this section, we perform Grad-CAM on Residual Networks (ResNets). In particular, we analyze the 200-layer architecture trained on ImageNet[2].

Current ResNets [40] typically consist of residual blocks. One set of blocks use identity skip connections (shortcut connections between two layers having identical output dimensions). These sets of residual blocks are interspersed with downsampling modules that alter dimensions of propagating signal. As can be seen in Fig. A.6 our visualizations applied on the last convolutional layer can correctly localize the cat and the dog. Grad-CAM can also visualize the cat and dog correctly in the residual blocks of the last set. However, as we go towards earlier sets of residual blocks with different spatial resolution, we see that Grad-CAM fails to localize the category of interest (see last row of Fig. A.6). We observe similar trends for other ResNet architectures (18 and 50-layer).

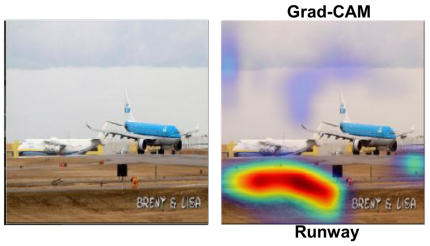
[2]We use the 200-layer ResNet architecture from `https://github.com/facebook/fb.resnet.torch`.



Important concepts for `**Canal Urban**`

Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
148	Water	338	Mountain
166	Water	244	Chequered
143	Water	184	Road
232	Sea	493	Grandstand
353	Boat	27	Airplane

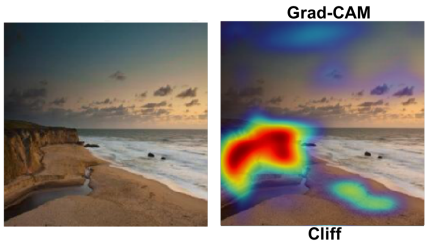
(a)



Important concepts for `**Runway**`

Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
262	Airplane	115	Balcony
500	Highway	166	Water
203	Airplane	494	Swimming Pool
498	Crosswalk	106	Closet
150	Airplane	412	Plant

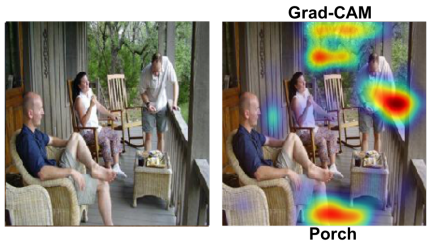
(b)



Important concepts for `**Cliff**`

Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
143	Water	488	Desert Sand
321	Mountain	299	Car
269	Tree	261	Snow
294	Pantry	404	Car
106	Closet	206	Car

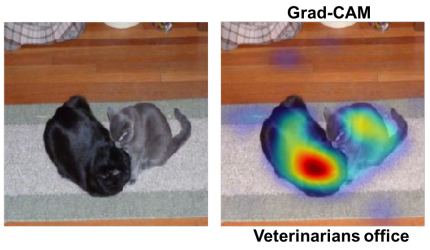
(c)



Important concepts for `**Porch**`

Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
146	Lined	375	Seat
28	Windowpane	304	Sink
373	Railing	258	Pool table
488	Desert sand	150	Airplane
338	Mountain	312	Desk

(d)

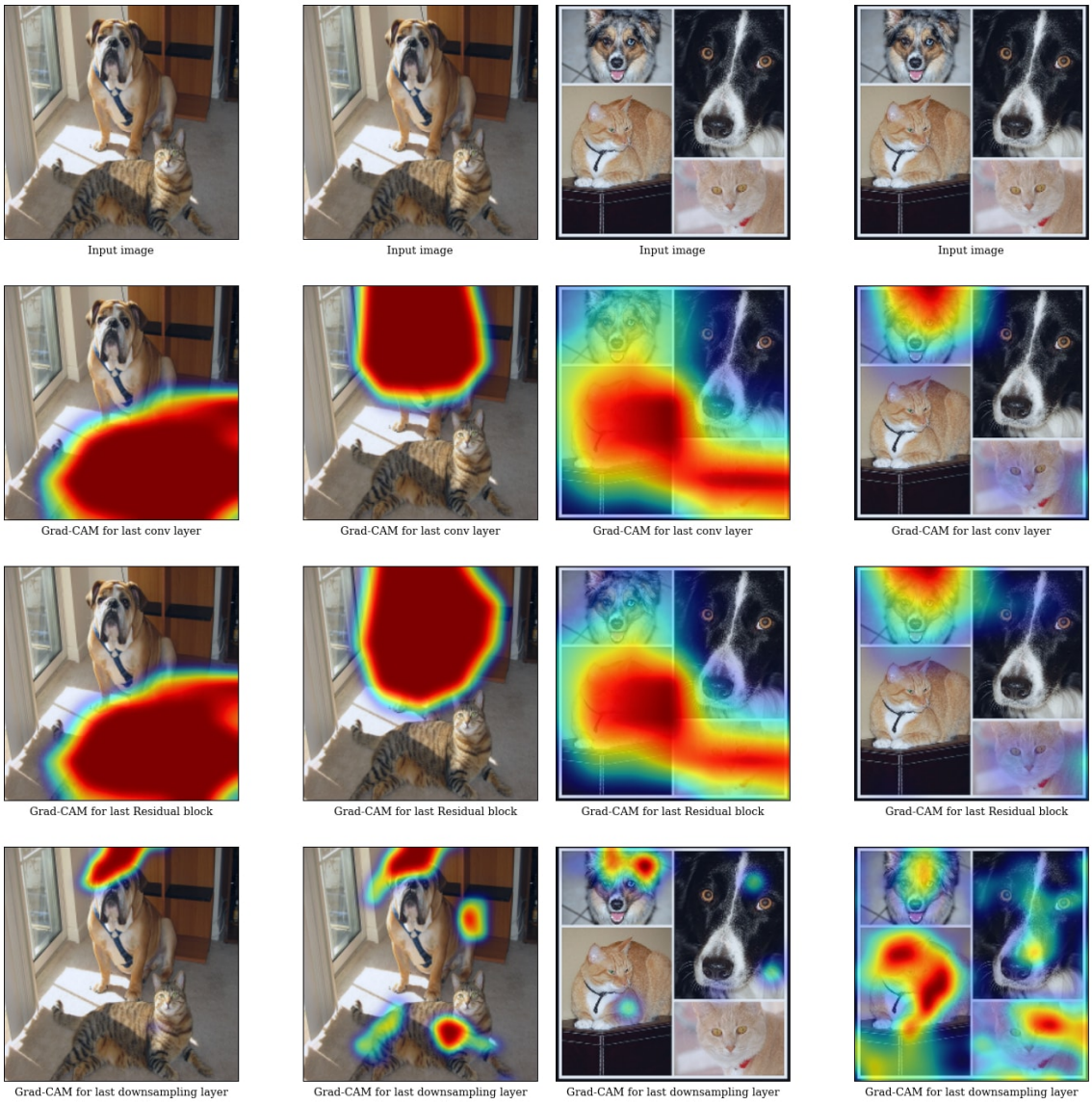


Important concepts for `**Veterinarians office**`

Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
141	Dog	406	Bakery shop
265	Head	504	Cracked
490	Dog	338	Mountain
324	Drawer	237	Sky
230	dotted	81	Cemetary

(e)

Figure A.5: More Qualitative examples showing visual explanations and textual explanations for VGG-16 trained on Places365 dataset ([68]). For textual explanations we provide the most important neurons for the predicted class along with their names. Important neurons can be either be persuasive (positively important) or inhibitive (negatively important).



(a) Grad-CAM visualizations for the ResNet-200 layer architecture for 'tiger cat'(left) and 'boxer'(right) category.

(b) Grad-CAM visualizations for the ResNet-200 layer architecture for 'tabby cat'(left) and 'boxer'(right) category.

Figure A.6: We observe that the discriminative ability of Grad-CAM significantly reduces as we encounter the downsampling layer.

APPENDIX B

APPENDIX FOR FACILITATING KNOWLEDGE TRANSFER BETWEEN HUMANS AND AI

B.1 Appendix Overview

In the appendix, we provide:

I - Details of finetuning the base model on seen classes

II - Results on SUN dataset

B.2 Finetuning on Seen Classes

We experiment with two particular convolutional architectures: ResNet101 [40] and VGG16 [60], pretrained on the ImageNet [57] dataset which is commonly used for pre-training deep classification networks for image classification. The CNNs were finetuned on the seen classes of the proposed split [86] for each of the datasets to obtain CNN classifiers for the same. The datasets CUB [100], AWA2 [86] and SUN [131] have 150, 40 and 645 seen classes respectively. Each of these datasets have certain number of images reserved for both seen and unseen classes to report generalized zero-shot learning performance in the proposed split. Excluding these images, we split the remaining images from the seen classes randomly into training (75%) and validation (25%) to finetune the CNNs. Finetuning was performed in two stages – first, only the last layer weights were trained with a cross-entropy loss with a fixed learning rate following which the weights of the entire network were updated with a (reduced) fixed learning rate. We used Adam [132] as the optimizer for all our experiments. We used early-stopping on the validation loss with a window of 20 epochs as our stopping criterion. We choose our hyper-parameters by grid search over the following ranges:

- `learning rate` (final layer) : $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$

- `learning rate` (all layers) : $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$
- `weight decay` : $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$

B.3 Results on SUN

Table B.1: Generalized Zero-Shot Learning performances on the proposed splits [86] for SUN [131]. We report class-normalized accuracies on seen and unseen classes and harmonic mean.[1] reproduced from [86]. [2] based on code provided by the authors. We see that NIWT is competitive with the best performing approaches on SUN.

			SUN [131]		
		Method	$\texttt{Acc}_{\mathcal{U}}$	$\texttt{Acc}_{\mathcal{S}}$	H
ResNet101 [40]	Fixed	ALE [80][1]	21.8	33.1	26.3
		SJE [82][1]	14.7	30.5	19.8
		DEVISE [81][1]	16.9	27.4	20.9
		CONSE [81][1]	6.8	39.9	11.6
VGG16 [60]	Fixed	Deep Embed. [102][2]	13.13	14.53	13.79
		NIWT-Attributes	**21.81**	**23.09**	**22.43**
	FT	Deep Embed. [102][2]	12.5	16.51	14.23
		NIWT-Attributes	**21**	**31.5**	**25.2**

We show results of NIWT on SUN [131] in Table. B.1 by using VGG16 [60] as the base architecture. We observe that NIWT performs competitively with the existing state-of-the-art methods on SUN across all the metrics - $\texttt{Acc}_{\mathcal{U}}$, $\texttt{Acc}_{\mathcal{S}}$ and H. It performs marginally worse compared to the best performing method on unseen classes – ALE [80] ($\sim 1\%$ worse on H and $\sim 0.8\%$ worse on $\texttt{Acc}_{\mathcal{U}}$) – when a finetuned version of VGG16 is used. Again, we observe that NIWT performs significantly better when fully-finetuned networks are used compared to Deep Embedding [102] which either reduces or maintains similar performance when going from frozen to fully fine-tuned features.

B.4 Qualitative examples

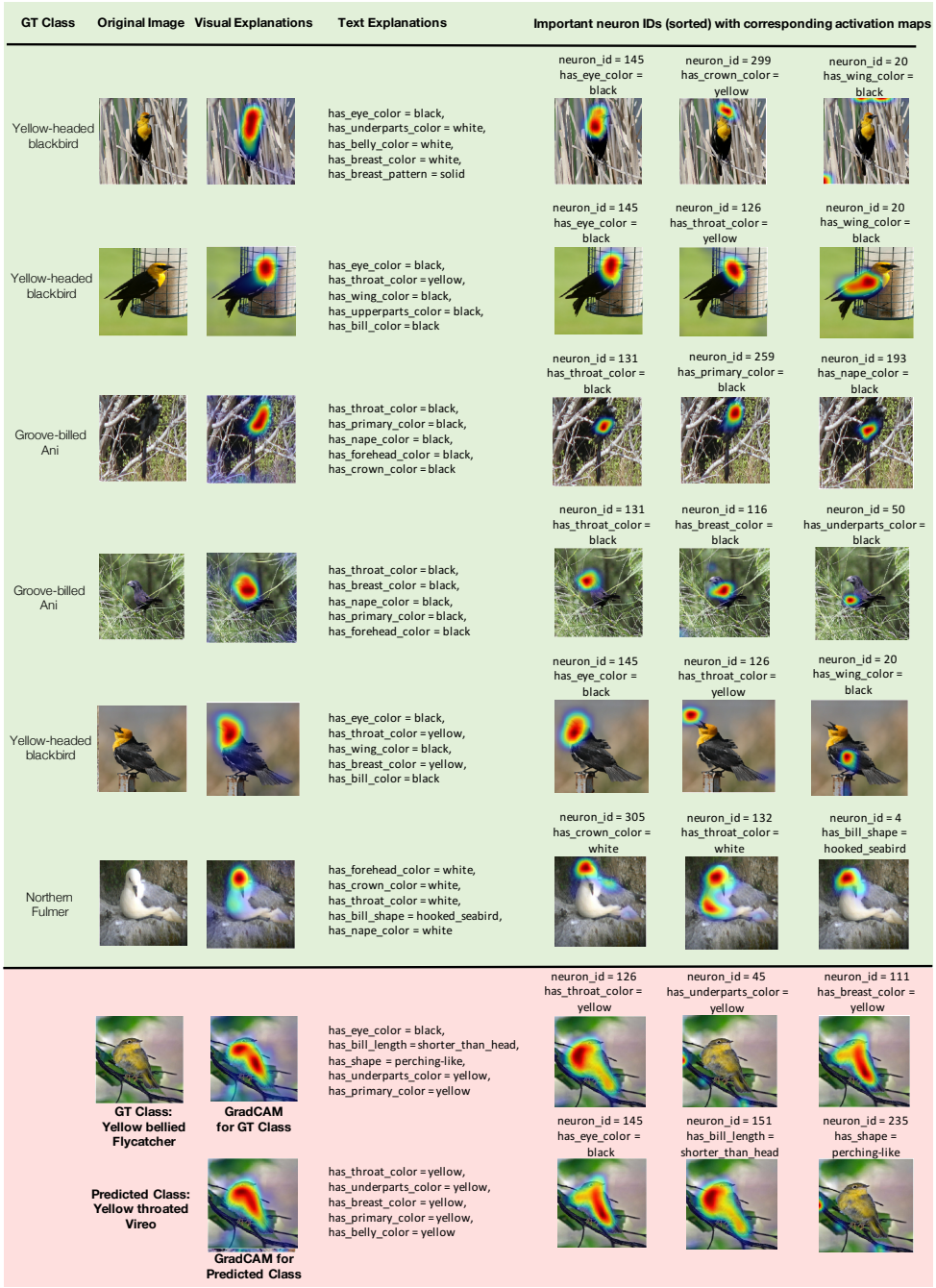


Figure B.1: Success and failure cases for unseen classes using explanations for NIWT: Success cases: (a) the ground truth class and image, (b) Grad-CAM visual explanations for the GT category, (c) textual explanations obtained using the inverse mapping from $\mathbf{a}_c$ to domain knowledge. (d) most important neurons for this decision and neuron names, including the activation map corresponding to the neuron. The last 2 rows show negative examples, where the model predicted a wrong category. We show Grad-CAM maps and textual explanations for both the ground truth and predicted category. By looking at the explanations for the failure cases we can see that the model's mistakes are not completely unreasonable.

APPENDIX C
APPENDIX FOR SQUINTING AT VQA MODELS

C.1 Introduction

We first provide a sample of the kind of regex-based rules that we used to arrive at reasoning questions. We then provide the interface we designed for training and evaluating Mechanical turk workers and the interface for collecting the main dataset. We then show randomly sampled responses from workers. We provide additional qualitative examples showing better grounding and improved task performance of SQuINTed models compared to the base model.

C.2 Perception-VQA vs Reasoning-VQA

In the first part of this section, we revisit our definition of Perception and Reasoning questions and later we describe our rules for constructing the Reasoning split.

C.2.1 Perception vs. Reasoning

Perception : As mentioned in Chapter 6, we define Perception questions as those which can be answered by detecting and recognizing the existence, physical properties and / or spatial relationships between entities, recognizing text / symbols, simple activities and / or counting, and that do not require more than one hop of reasoning or general commonsense knowledge beyond what is visually present in the image. Some examples are: "Is that a cat? " (existence), "Is the ball shiny?" (physical property), "What is next to the table?" (spatial relationship), "What does the sign say?" (text / symbol recognition), "Are the people looking at the camera?" (simple activity), etc.

Reasoning : We define Reasoning questions as non-Perception questions which require the

synthesis of perception with prior knowledge and / or reasoning in order to be answered. For instance, “Is this room finished or being built?”, “At what time of the day would this meal be served?”, “Does this water look fresh enough to drink?”, “Is this a home or a hotel?”, “Are the giraffes in their natural habitat?” are all Reasoning questions.

C.2.2 Rules

As mentioned in Chapter 6, our analysis of the perception questions in the VQA dataset revealed that most perception questions have distinct patterns that can be identified with high precision regex-based rules. In Table C.1 we provide a list of top-40 regex rules based on the percentage of data the rule eliminated.

By hand-crafting such rules (as seen in Table C.1) and filtering out perception questions, we identify 18% of the VQA dataset as highly likely to be Reasoning.

C.2.3 Validating rules

To check the accuracy of our rules, we designed a crowdsourcing task on Mechanical Turk that instructed workers to identify a given VQA question as Perception or Reasoning, and to subsequently provide sub-questions for the Reasoning questions.

Validating Precision. As mentioned in Section 3.1, 94.7% of the times, trained workers classified our resulting questions as reasoning questions demonstrating the high precision of the regex-based rules we created.

C.3 Sub-VQA

In this section, we describe how we collect sub-questions and answers for questions in our Reasoning split.

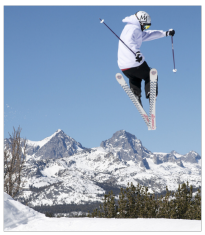
Given the complexity of distinguishing between Perception / Reasoning and providing sub-questions for Reasoning questions, we first train and filter workers on Amazon

Main Reasoning Question:
- Is this at a residence or restaurant? "Restaurant"

Perception Sub-questions:
- Does the napkin have a name of a cafe or a restaurant written on it? "Yes"

Main Reasoning Question:
- Has the pizza cutter been used yet? "No"

Perception Sub-questions:
- Is the pizza cut? "no"
- Is the pizza cut up yet, or one piece? "One piece"



Main Reasoning Question:
- Is he going to land safely? "Yes"

Perception Sub-questions:
- Are the skis pointed toward the ground? "Yes"
- Is the person facing the direction that they are falling? "Yes"
- Are the skis below the persons body and above the ground? "Yes"
- Is the person in the air? "Yes"

Main Reasoning Question:
- Is this a happy couple? "Yes"

Perception Sub-questions:
- Are the people smiling? "Yes"
- Is the bride smiling? "Yes"
- Did the couple just get married? "Yes"

Main Reasoning Question:
- Would it be safe to suggest most of the vegetation shown would not hide this animal? "Yes"

Perception Sub-questions:
- What kind of animal is this? "Elephant"
- Is the elephant larger than the tree trunks? "Yes"
- What kind of plants are around the elephant? "Trees"

Main Reasoning Question:
- Could you pick up this pizza to eat it? "No"

Perception Sub-questions:
- Is the sauce thick and running down the side? "Yes"
- Is the topping heavy, loose with lots of sauce? "Yes"
- Is there a very thick slice of pizza? "Yes"

Figure C.1: Randomly sampled qualitative examples of Perception sub-questions in our VQA-Introspect dataset for main questions in the Reasoning split of VQA. Main questions are written in orange and sub questions are in blue. A single worker may have provided more than one sub questions for the same (image, main question) pair.

Mechanical Turk (AMT) via qualification rounds before we rely on them to generate high-quality sub-questions.

Worker Training - We manually annotate 100 questions from the VQA dataset as Perception and 100 as Reasoning questions, to serve as examples. We first teach workers the difference between Perception and Reasoning questions by defining them and showing several examples of each, along with explanations. Then, workers are shown (question, answer) pairs and are asked to identify if the given question is a Perception question or a Reasoning question [1]. Finally, for Reasoning questions, we ask workers to add all Perception questions and corresponding answers (in short) that would be necessary to answer the main question. In this qualification HIT, workers have to make 6 Perception and Reasoning judgments, and they qualify if they get 5 or more answers right. This interface can be found under the name *'qual1_interace.html'* in attached zip file.

We launched further pilot experiments for the workers who passed the first qualification round, where we manually evaluated the quality of their sub-questions based on 2 criteria : (1) The sub-questions should be Perception questions grounded in the image, and 2) The sub-questions should be sufficient to answer the main Reasoning question. Among those 463 workers who passed the first qualification test, 91 were selected (via manual evaluation) as high-quality workers, which finally qualified for attempting our main task.

Main task - In the main data collection, all VQA questions that got identified as Reasoning by regex-rules (section C.2) and a random subset of the questions identified as Perception were further judged by workers (for validation purposes). We eliminated ambiguous questions by further filtering out questions where there is high worker disagreement about the answer. We require at least 8 out of 10 workers to agree with the majority answer for yes/no questions and 5 out of 10 for all other questions, which leaves us with a split that corresponds to $\sim$13% of the VQA dataset. This interface can be found under the name *'main_interace.html'* in attached zip file.

[1] We also add an "Invalid" category to flag nonsensical questions or those which can be answered without looking at the image

Until the time of submission, we have collected sub questions for VQAv1 train which corresponds to 27441 Reasoning questions and 79905 sub questions for them. For VQAv2 val we have 15448 Reasoning questions and 52573 sub questions for them.

C.4 VQA-Introspect

Each <question, image> pair labeled as Reasoning had sub questions generated by by 3 unique workers [2]. On average we have 2.60 sub-questions per Reasoning question.

Randomly sampled qualitative examples from our collected dataset are shown in Fig. C.2.

C.5 SQuINT Qualitative results

In Fig. C.3 we show more qualitative examples showing the effect of applying SQuINT to the Pythia model on a held out val set.

[2] A small number of workers displayed degraded performance after the qualification round, and were manually filtered

Table C.1: Our top-40 rules for eliminating perception questions. Length refers to the words in the question.

Rules				Amount of Data	
Starts with	Contains	Not contains	Length	# questions	% data
How many	-	-	-	48656	10.96
-	color	-	-	47956	10.81
What is the	-	-	-	40988	9.24
What	on	-	-	29031	6.54
What	in	-	-	21876	4.93
Is there	-	-	-	16494	3.72
-	wear	['appropriate', 'acceptable', 'etiquitte']	-	15530	3.50
-	wearing	-	-	14940	3.37
Is this a	-	-	4	14814	3.34
Where	-	-	-	12409	2.80
-	old	-	-	11197	2.52
What kind of	-	-	-	11186	2.52
What are	-	-	-	10524	2.37
-	on?	-	-	9040	2.04
Are there	-	-	-	8665	1.95
What type of	-	-	-	7955	1.79
-	doing?	-	-	7288	1.64
-	holding	-	-	7137	1.61
-	low	-	-	6596	1.49
-	round?	-	-	6242	1.41
Do	have	-	-	6213	1.40
Is the	on the	-	-	5375	1.21
Are these	-	['homemade', 'healthy', 'domesticated', etc.]	3	5320	1.20
Is the	in the	['wild', 'mountain', 'desert', 'woods', etc.]	-	5108	1.15
Does	have	-	-	5078	1.14
-	number	-	-	4477	1.01
What is this	-	-	-	3970	0.89
Is	ed?	['overexposed?', 'doctored?', 'ventilated?', etc.]	3	3940	0.88
Is	ing?	['horrifying?', 'relaxing?', 'competing?', etc.]	3	3870	0.88
Is	on	-	3	3622	0.82
Who	on	-	-	3563	0.80
-	shown?	-	-	3501	0.79
What sport	-	-	-	3412	0.77
-	sun	-	-	3260	0.73
-	see	-	-	3238	0.73
-	visible	-	-	3076	0.69
What	say?	-	-	3238	0.69
What	playing?	-	-	3076	0.69
Are the	in the	['US', 'wild', 'team', 'or', etc.]	-	3010	0.68
What	playing?	-	-	3076	0.69
Are	on the	-	-	2932	0.66

Main Reasoning Question:
- Are the animals in their natural habitat? “Yes”

Perception Sub-questions:
- Is the bear touching a log in the water? “Yes”
- Does the bear have a wild water source to thrive in? “Yes”
- Is the bear in water? “Yes”
- Is the bear in a zoo? “No”
- Is the bear caged or fenced in? “No”

Main Reasoning Question:
- Is this a vegan dish? “Yes”

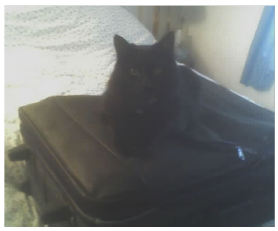
Perception Sub-questions:
- Is there any meat on the dish or dairy? “No”
- Are only vegetables shown? “Yes”
- Is there meat or dairy on the plate? “No”

Main Reasoning Question:
- Was this picture taken in Australia? “Yes”

Perception Sub-questions:
- What type of animals are climbing the tree? “small bears”
- Are there koala bears in the tree? “Yes”

Main Reasoning Question:
- Do you need practice to use one of these? “Yes”

Perception Sub-questions:
- Are the person's feet on a snowboard? “Yes”
- What is the object the person is standing on? “Snowboard”
- Has a person fallen? “Yes”



Main Reasoning Question:
- Is the cat a tabby? “No”

Perception Sub-questions:
- What is the color of the cat? “Black”
- Is the cat orange? “No”
- What color is the cat? “Yes”

Main Reasoning Question:
- Was this taken during the day? “No”

Perception Sub-questions:
- Is it dark outside? “Yes”

Main Reasoning Question:
- Could this be a foreign country? “Yes”

Perception Sub-questions:
- What type of vehicles are on the street? “Scooters”
- Are there people riding a motorbike? “Yes”
- Are the helmets the style worn in America? “No”

Figure C.2: More randomly sampled qualitative examples of Perception sub-questions in our VQA-Introspect dataset for main questions in the Reasoning split of VQA.

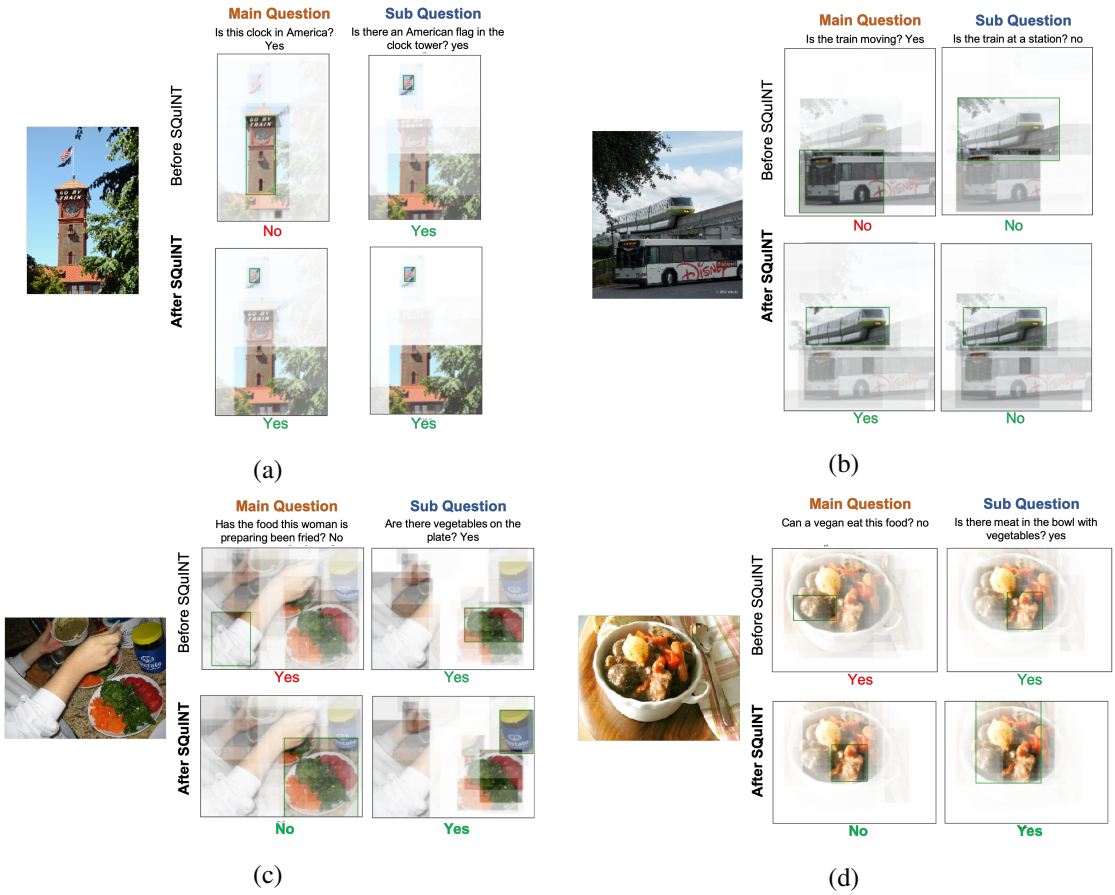


Figure C.3: Qualitative examples showing the model attention before and after applying SQuINT. (a) shows an image along with the reasoning question, *'Is this clock in America?'*, for which the Pythia model looks at the tower regions and answers "No" incorrectly. Note how the same model correctly looks at the flag above to answer the easier sub-question, *'Is there an American flag in the clock tower?'*. After applying SQuINT, which encourages the model to use the perception based sub question attention while answering the reasoning question, now looks at the flag and correctly answers the main reasoning question.

REFERENCES

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C Lawrence Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *ICCV*, 2015.

[2] A. Agrawal, D. Batra, and D. Parikh, "Analyzing the behavior of visual question answering models," in *EMNLP*, 2016.

[3] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing Error in Object Detectors," in *ECCV*, 2012.

[4] A. Karpathy, *What I learned from competing against a ConvNet on ImageNet*, http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/, 2014.

[5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[6] E. Johns, O. Mac Aodha, and G. J. Brostow, "Becoming the Expert - Interactive Multi-Class Machine Teaching," in *CVPR*, 2015.

[7] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, "Don't just assume; look and answer: Overcoming priors for visual question answering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[8] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2013.

[9] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for Simplicity: The All Convolutional Net," *CoRR*, vol. abs/1412.6806, 2014.

[10] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[11] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "Devnet: A deep event network for multimedia event detection and evidence recounting," in *CVPR*, 2015.

[12] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing Higher-layer Features of a Deep Network," *University of Montreal*, vol. 1341, 2009.

[13] A. Mahendran and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," *International Journal of Computer Vision*, pp. 1–23, 2016.

[14] A. Dosovitskiy and T. Brox, "Inverting Convolutional Networks with Convolutional Networks," in *CVPR*, 2015.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *SIGKDD*, 2016.

[16] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization.," in *CVPR*, 2016.

[18] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2014.

[19] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, Jan. 2009.

[20] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 951–958.

[21] D. Parikh and K. Grauman, "Relative attributes," in *2011 International Conference on Computer Vision*, IEEE, 2011, pp. 503–510.

[22] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 819–826.

[23] S. J. Hwang and L. Sigal, "A unified semantic embedding: Relating taxonomies and attributes," in *Advances in Neural Information Processing Systems*, 2014, pp. 271–279.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[25] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *International conference on machine learning*, 2016, pp. 2397–2406.

[27] V. Kazemi and A. Elqursh, "Show, ask, attend, and answer: A strong baseline for visual question answering," *CoRR*, vol. abs/1704.03162, 2017.

[28] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 21–29.

[29] J. Lu, J. Yang, D. Batra, and D. Parikh, "Neural baby talk," in *CVPR*, 2018.

[30] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018.

[31] R. R. Selvaraju, S. Lee, Y. Shen, H. Jin, S. Ghosh, L. Heck, D. Batra, and D. Parikh, "Taking a hint: Leveraging explanations to make vision and language models more grounded," in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[32] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, "Yin and Yang: Balancing and answering binary visual questions," in *CVPR*, 2016.

[33] L. Anne Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach, "Women also snowboard: Overcoming bias in captioning models," in *ECCV*, 2018.

[34] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *CVPR*, 2017.

[35] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, "Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions?" In *EMNLP*, 2016.

[36] D. Huk Park, L. Anne Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach, "Multimodal explanations: Justifying decisions and pointing to the evidence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8779–8788.

[37] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization.," in *ICCV*, 2017.

[38] T. Qiao, J. Dong, and D. Xu, "Exploring human-like attention supervision in visual question answering," in *AAAI*, 2018.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *CVPR*, 2014.

[42] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[43] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015.

[44] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft COCO captions: Data Collection and Evaluation Server," *arXiv preprint arXiv:1504.00325*, 2015.

[45] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, *et al.*, "From Captions to Visual Concepts and Back," in *CVPR*, 2015.

[46] J. Johnson, A. Karpathy, and L. Fei-Fei, "DenseCap: Fully Convolutional Localization Networks for Dense Captioning," in *CVPR*, 2016.

[47] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, "Are you talking to a machine? dataset and methods for multilingual image question," in *Advances in Neural Information Processing Systems*, 2015, pp. 2296–2304.

[48] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *ICCV*, 2015.

[49] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *NIPS*, 2015.

[50] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual Dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[51] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. C. Courville, "Guesswhat?! visual object discovery through multi-modal dialogue," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[52] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra, "Learning cooperative visual dialog agents with deep reinforcement learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[53] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied Question Answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "Iqa: Visual question answering in interactive environments," *arXiv preprint arXiv:1712.03316*, 2017.

[55] Z. C. Lipton, "The Mythos of Model Interpretability," *ArXiv e-prints*, Jun. 2016. arXiv: 1606.03490 [cs.LG].

[56] P. Jackson, *Introduction to Expert Systems*, 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998, ISBN: 0201876868.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[58] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Computer Vision and Pattern Recognition*, 2017.

[59] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[60] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR*, 2015.

[61] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[62] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *ACMMM*, 2014.

[63] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down Neural Attention by Excitation Backprop," in *ECCV*, 2016.

[64] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *ECCV*, 2016.

[65] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization," *CoRR*, vol. abs/1610.02391, 2016.

[66] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "HOGgles: Visualizing Object Detection Features," *ICCV*, 2013.

[67] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *stat*, 2015.

[68] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[69] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *CoRR*, vol. abs/1412.6856, 2014.

[70] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Computer Vision and Pattern Recognition*, 2017.

[71] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[72] J. Lu, X. Lin, D. Batra, and D. Parikh, *Deeper LSTM and normalized CNN Visual Question Answering model*, `https://github.com/VT-vision-lab/VQA_LSTM_CNN`, 2015.

[73] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *NIPS*, 2016.

[74] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, "Stacked attention networks for image question answering," *CoRR*, vol. abs/1511.02274, 2015.

[75] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks.," in *AAAI*, vol. 1, 2008, p. 3.

[76] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings," in *ICLR*, 2014.

[77] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in neural information processing systems*, 2013.

[78] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[79] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[80] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1425–1438, 2016.

[81] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *Advances in neural information processing systems*, 2013.

[82] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2927–2936.

[83] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 2152–2161.

[84] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *CVPR*, 2016.

[85] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[86] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning - the good, the bad and the ugly," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[87] R. Qiao, L. Liu, C. Shen, and A. v. d. Hengel, "Less is more: Zero-shot learning from online textual documents with noise suppression," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, year=2016*.

[88] M. Elhoseiny, B. Saleh, and A. Elgammal, "Write a classifier: Zero-shot learning using purely textual descriptions," in *The IEEE International Conference on Computer Vision (ICCV)*, 2013.

[89] M. Elhoseiny, Y. Zhu, H. Zhang, and A. Elgammal, "Link the head to the "beak": Zero shot learning from noisy text description at part precision," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[90] M. Elhoseiny, A. Elgammal, and B. Saleh, "Write a classifier: Predicting visual classifiers from unstructured text," *Ieee T Pattern Anal*, vol. PP, no. 99, pp. 1–1, 2017.

[91] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong, "Transductive multi-view embedding for zero-shot recognition and annotation," in *European Conference on Computer Vision*, Springer, 2014, pp. 584–599.

[92] S. Huang, M. Elhoseiny, A. Elgammal, and D. Yang, "Learning hypergraph-regularized attribute predictors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 409–417.

[93] J. Lei Ba, K. Swersky, S. Fidler, *et al.*, "Predicting deep zero-shot convolutional neural networks using textual descriptions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4247–4255.

[94] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 49–58.

[95] R. Yu, A. Li, C. Chen, J. Lai, V. I. Morariu, X. Han, M. Gao, C. Lin, and L. S. Davis, "NISP: pruning networks using neuron importance score propagation," *CVPR*, 2018.

[96] S. Konam, I. Quah, S. Rosenthal, and M. Veloso, "Understanding convolutional networks with apple : Automatic patch pattern labeling for explanation," *First AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[97] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[98] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814.

[99] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and generalization in neural networks: An empirical study," *arXiv preprint arXiv:1802.08760*, 2018.

[100] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[101] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[102] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *CVPR*, 2017.

[103] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *CVPR*, 2017.

[104] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*, 2017, pp. 3319–3328.

[105] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[106] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization.," *International Journal of Computer Vision*, 2019.

[107] C. Liu, J. Mao, F. Sha, and A. L. Yuille, "Attention correctness in neural image captioning.," in *AAAI*, 2017.

[108] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *ECCV*, 2014.

[109] Y. Goyal, A. Mohapatra, D. Parikh, and D. Batra, "Interpreting visual question answering models," *CoRR*, vol. abs/1608.08974, 2016. arXiv: 1608.08974.

[110] S. Ramakrishnan, A. Agrawal, and S. Lee, "Overcoming language priors in visual question answering with adversarial regularization," in *Neural Information Processing Systems (NIPS)*, 2018.

[111] R. R. Selvaraju, P. Chattopadhyay, M. Elhoseiny, T. Sharma, D. Batra, D. Parikh, and S. Lee, "Choose your neuron: Incorporating domain knowledge through neuron-importance," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 526–541.

[112] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[113] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4651–4659.

[114] Z. Y. Y. Y. Y. Wu and R. S. W. W. Cohen, "Encode, review, and decode: Reviewer module for caption generation," *arXiv preprint arXiv:1605.07912*, 2016.

[115] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 6, 2017, p. 2.

[116] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[117] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014.

[118] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, "Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions?" In *EMNLP*, 2016.

[119] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[120] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," *CoRR*, vol. abs/1411.5726, 2014.

[121] D. D. Hoffman and W. A. Richards, "Parts of recognition," *Cognition*, vol. 18, no. 1-3, pp. 65–96, 1984.

[122] J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis," *Cognition*, vol. 28, no. 1-2, pp. 3–71, 1988.

[123] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *Advances in Neural Information Processing Systems*, 2015, pp. 2953–2961.

[124] M. Shah, X. Chen, M. Rohrbach, and D. Parikh, "Cycle-consistency for robust visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6649–6658.

[125] M. T. Ribeiro, C. Guestrin, and S. Singh, "Are red roses red? evaluating consistency of question-answering models," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6174–6184.

[126] D. A. Hudson and C. D. Manning, "GQA: A new dataset for real-world visual reasoning and compositional question answering," in *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[127] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, "Pythia v0. 1: The winning entry to the vqa challenge 2018," *arXiv preprint arXiv:1807.09956*, 2018.

[128] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, Elsevier, 1989, pp. 109–165.

[129] D. A. Hudson and C. D. Manning, "Compositional attention networks for machine reasoning," *arXiv preprint arXiv:1803.03067*, 2018.

[130] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2009.

[131] G. Patterson, C. Xu, H. Su, and J. Hays, "The sun attribute database: Beyond categories for deeper scene understanding," *International Journal of Computer Vision*, vol. 108, no. 1-2, pp. 59–81, 2014.

[132] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

VITA

Ramprasaath is a Computer Science PhD student at Georgia Institute of Technology, advised by Devi Parikh, and working closely with Dhruv Batra. He works at the intersection of machine learning, computer vision & language and explainable AI. Specifically, his research focuses on building algorithms that provide explanations for decisions emanating from deep networks in order to build user trust, incorporate domain knowledge into AI, and correct for unwanted biases learned by deep AI models. He has previously interned at Brown University (Summer 2013), Oxford University (Fall 2014), Facebook (Spring 2017), Samsung Research (Summer 2018), Tesla Autopilot (Spring 2019) and Microsoft Research (Summer 2019). He graduated from Birla Institute of Technology and Science in 2015 with a Bachelor's degree in Electrical and Electronics Engineering and a Masters in Physics.